KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF SCIENCE

Simulation of wave propagation problems for automated characterisation of material parameters

Steven Vandekerckhove

Supervisors:
Prof. dr. ir. H. De Gersem
Prof. dr. K. Van Den Abeele
Prof. dr. ir. S. Vandewalle

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Science (PhD): Physics

September 2016

Simulation of wave propagation problems for automated characterisation of material parameters

Steven VANDEKERCKHOVE

Examination committee:
Prof. dr. P. Igodt, chair
Prof. dr. ir. H. De Gersem, supervisor
Prof. dr. K. Van Den Abeele, supervisor
Prof. dr. ir. S. Vandewalle, supervisor
Prof. dr. ir. L. De Lathauwer
Prof. dr. ir. D. Huybrechts
Prof. dr. Garth N. Wells
(University of Cambridge)
dr. Patrick E. Farrell
(University of Oxford)

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Science (PhD): Physics

September 2016

Uitgegeven in eigen beheer, Steven Vandekerckhove, E. Sabbelaan 53, B-8500 Kortrijk (Belgium)

Dankwoord

Beste promotoren, gedurende de 1825 dagen dat ik jullie doctoraatsstudent was, hebben we elkaar gemiddeld 1805 dagen (+/- 5 dagen) niet gesproken over mijn onderzoek. Nu blijkt dit de ideale strategie geweest te zijn om van mij een zelfstandige onderzoeker te maken, waarvoor ik jullie graag wil bedanken. Daarnaast heb ik nog een lange lijst met dingen waarvoor ik jullie wil bedanken. Ik beperk me hier tot enkele elementen op die lijst, maar ik overloop met veel plezier de volledige lijst tijdens een goed gesprek... Bedankt voor de vele waardevolle levenslessen die jullie aan mij doorgaven. Bedankt om me mijn eigen koers te laten varen. Bedankt om op het juiste moment de juiste dingen te zeggen. Bedankt…

Ik heb geleerd dat mensen zoeken die je kunnen en willen helpen een belangrijke (zoniet de belangrijkste) vaardigheid is van een zelfstandige onderzoeker. Aangezien ik dat dan ook uitvoerig gedaan heb, grijp ik hier met veel plezier de kans om nog heel wat mensen te bedanken op wie ik een beroep heb mogen doen om mijn doctoraatsopleiding tot een goed einde te brengen.

Garth and Patrick, you are the first people I would like to thank next, together with the other members of the FEniCS community. It was always a pleasure to meet with you, discuss with you and learn from you. Without your input, large parts of this thesis would not be what they are today. Whenever I couldn't find an answer to a problem I was facing in Kortrijk or in Leuven, you welcomed me in Cambridge and Oxford. I have many nice memories and obtained valuable knowledge during my visits to you both.

Naast de FEniCS community had ik ook het plezier om me thuis te voelen op activiteiten van de Werkgroep Scientific Computing en de workshops in Kleinwalsertal. Aan de vele mensen die me op deze, maar ook de vele andere internationale bijeenkomsten die ik bijwoonde inspireerden, op welke manier dan ook, zeg ik oprecht dankuwel. Dat herinnert mij er aan dat ik ook mijn jobstudenten wil bedanken. Michiel, Febe en Hannes, dankjewel om ervoor te zorgen dat mijn onderzoek niet stil stond terwijl ik op conferentie was. Het was een plezier om met elk van jullie samen te werken en jullie te zien groeien van implementatieslaafjes tot collega's die een waardevolle bijdrage geleverd hebben aan mijn onderzoek.

Beste leden van mijn jury, ik wil jullie graag bedanken voor het grondig nalezen van deze tekst, de daaropvolgende suggesties en voor jullie interesse in mijn onderzoek die duidelijk voelbaar was tijdens de verschillende evaluatiemomenten.

Ik wil ook graag de KU Leuven en het IWT bedanken. Zonder hun financiële steun zou deze tekst niet bestaan.

Tijdens en naast het werken op de Kulak heb ik heel veel leuke momenten beleefd die ik te danken heb aan de goede sfeer onder de collega's. Erik, Dries, Maarten en Gert-Jan, jullie wil ik in het bijzonder bedanken voor o.a. de vele bordspelavonden en bezoekjes aan het wetenschapscafé. Steven en Eef, doctor in o.a. de knutselwetenschapen, dankjewel voor de aangename samenwerking en het doorgeven van jullie jarenlange ervaring.

Naast mijn onderzoek heb ik ook geijverd om de stem van de doctorandi op de Kulak wat luider te laten klinken. Met als resultaat dat er een werkgroep van en voor doctorandi werd opgericht op de Kulak. Ik wil mijn collega's bedanken die hier samen met mij hun tijd aan besteedden. Zonder jullie zou ik de Kulak verlaten zonder de waardevolle ervaring om in team een gemeenschappelijk doel na te streven.

Liefste familie, ik ben trots op jullie, om de indrukwekkende dingen die we steeds weer bereiken door elkaar te helpen en samen te werken. De voorbije vijf jaar was ik niet altijd even toegankelijk omdat ik bijvoorbeeld op een onderdeel van mijn onderzoek aan het kauwen was of voor de zoveelste keer op conferentie was. Hartelijk dank voor jullie steun, begrip en geduld! Mams en paps, bedankt voor de vele duwtjes in rug op de lange weg richting het verdedigen van dit werk en het helpen relativeren van de frustraties die ik op die weg tegenkwam.

Bedankt (en sorry) aan de vele mensen die ik niet bij naam genoemd heb, weet dat dat enkel een kwestie van plaatsgebrek is.

Nonkel Jo, jij zorgde er meer dan 15 jaar geleden voor dat ik voor het eerst hoorde over de boeiende dingen die je kan doen met een opleiding in de wetenschappen en de vaardigheden die je tijdens een doctoraatsopleiding kan ontwikkelen. Aangezien jij mij het eerste duwtje richting doctoraat gaf, en ik daar nog geen moment spijt van heb gehad, is mijn laatste dankjewel in dit voorwoord voor jou.

Steven

Abstract

In many application fields, non-destructive testing and evaluation (NDT&E) is useful to determine material properties and detect faults of objects before or during operation. In the past decades, a wide range of experimental NDT&E techniques has been developed and successfully tested for many problem classes. In the past few years, the collection of experimental NDT&E techniques has been extended with hybrid methods, i.e., methods that combine experiments and numerical simulation. The main drawback of all existing NDT&E methods, is that they need a-priori knowledge and expertise of the operator. In this work, we aim to develop automated procedures for the characterisation of material parameters based on the simulation of wave propagation problems, thereby extending the application range of NDT&E, especially to problems with little a-priori knowledge.

We propose the use of gradient based optimisation methods to develop automated procedures for the characterisation of material parameters. To achieve this, we first select efficient numerical procedures for the solution of wave propagation problems. For the spatial discretisation, we consider the finite integration technique and the finite element method with both continuous and discontinuous elements. We study the performance of several time integration methods in combination with these spatial discretisation methods, from which we learn that third order discontinuous elements in combination with second order time stepping methods are performing best.

We compute gradient information of the time-dependent wave propagation problem and use this in combination with the forward solver to efficiently solve the inverse problems that occur in NDT&E. For the efficient computation of gradients, adjoint methods are used. The proposed gradient based optimisation method is compared to gradient free methods for determining the wave speeds in a homogeneous isotropic medium, from which we conclude that gradient based methods are the only feasible option to tackle more complicated inverse problems. The challenges involved in solving optimisation problems with spatially dependent control parameters are described and successfully demonstrated for determining a spatially dependent wave speed distribution. We show that the use of proper function spaces and associated Riesz maps is a key ingredient to obtain correct results.

We also use gradient information to automatically calibrate absorbing layers to mimic open boundaries. The automatic calibration is performed for perfectly matched layers and for the simpler and computationally faster approach of using consecutive absorbing layers. We show that after calibration, both approaches have a comparable performance.

Throughout this work, we use a high level of genericity to describe and simulate wave propagation problems so that all presented methods are applicable to acoustic, electromagnetic and elastic wave propagation problems in one two and three dimensions. This work demonstrates that numerically solving inverse NDT&E problems with many, e.g. spatially dependent, parameters is possible.

Beknopte samenvatting

In veel toepassingsgebieden is niet-destructief testen en evalueren (NDT&E) een handige techniek om materiaaleigenschappen te bepalen en defecten te detecteren voor of tijdens het gebruik van voorwerpen. In de voorbije decennia is een uitgebreid bereik aan experimentele NDT&E technieken ontwikkeld en succesvol toegepast op veel verschillende soorten problemen. In de voorbije jaren is het arsenaal aan experimentele NDT&E technieken uitgebreid met hybride methodes, dit zijn methodes die experimenten en numerieke simulaties combineren. Het grootste nadeel van de bestaande NDT&E methodes, is dat de gebruiker moet beschikken over heel wat voorkennis en expertise. Het objectief van dit werk is om automatische procedures te ontwikkelen voor het bepalen van materiaalparameters gebaseerd op simulatie van golfvoortplantingsproblemen. Hierdoor breiden we het toepassingsbereik van NDT&E uit tot problemen met beperkte voorkennis.

Wij stellen voor om gradiënt gebaseerde optimalisatiemethodes te gebruiken om automatische procedures te ontwikkelen voor het bepalen van materiaalparameters. Om dit te bereiken selecteren we eerst efficiënte methodes voor de numerieke simulatie van golfvoortplantingsproblemen. Voor de ruimtelijke discretisatie beschouwen we de eindige integratie techniek alsook de eindige elementen methode met zowel continue als discontinue elementen. We bestuderen de performantie van verschillende tijdintegratoren in combinatie met deze ruimtelijke discretisatietechnieken. Hieruit leren we dat het gebruik van derde orde discontinue elementen in combinatie met tweede orde tijdsintegratoren tot de beste performantie leidt.

We berekenen gradiëntinformatie van het tijdafhankelijke golfvoortplantingsprobleem en gebruiken deze informatie in combinatie met de oplossingsmethodes voor voorwaartse golfvoortplantingsproblemen om inverse problemen die zich voordoen in NDT&E op te lossen. Voor de efficiënte berekening van de gradiënten, wordt gebruik gemaakt van *adjoint* methodes. De voorgestelde gradiënt gebaseerde optimalisatie methode wordt vergeleken met gradiëntvrije

methodes voor het bepalen van golfsnelheden in homogene isotrope media. Uit deze studie besluiten we dat enkel gradiënt gebaseerde methodes een haalbare optie zijn voor het oplossen van ingewikkeldere inverse problemen. De uitdagingen die zich voordoen met ruimtelijk afhankelijke controleparameters worden beschreven en succesvol gedemonstreerd om een ruimtelijk afhankelijke golfsnelheid te bepalen. Wij tonen aan dat het gebruik van goedgekozen functieruimtes en daarbij horende Rieszafbeeldingen essentieel is om zinvolle resultaten te bekomen.

We gebruiken de gradiëntinformatie ook om op automatische wijze absorberende lagen die een open randvoorwaarde nabootsen te kalibreren. De automatische kalibratie wordt gebruikt voor *perfectly matched layers* alsook voor eenvoudige opeenvolgende gedempte absorberende lagen. We tonen aan dat na kalibratie, beide aanpakken een vergelijkbare performantie hebben.

Doorheen dit werk maken we gebruik van een generieke aanpak voor de beschrijving en berekening van golfvoortplantingsproblemen zodat alle voorgestelde technieken gebruikt kunnen worden voor akoestische, elektromagnetische en elastische golfvoortplantingsproblemen in één, twee en drie dimensies. In dit werk tonen we aan dat het mogelijk is om inverse NDT&E problemen met veel parameters, zoals ruimtelijke afhankelijke problemen, op te lossen.

Contents

Chapter 1

Introduction

1.1 Motivation

To make life as we know it sustainable for an increasing number of people on the planet, we need more efficient production processes, more eco-friendly transportation, cleaner energy production, optimal resource usage, products that are easier to recycle, etc. In this quest for sustainability, a battle between economical and ecological interests is going on. These interests often appear to be opposing, but are not always. For example, when less aluminium is used to make the chassis of a vehicle, then the final product will be lighter, which will reduce production cost and fuel consumption. A lighter chassis might cause the vehicle to be less robust which may lead to reduced safety. There are however measures that can compensate for the reduced safety due to the loss of robustness, such as an optimised chassis shape or monitoring chassis integrity. Non-destructive testing (NDT) of material integrity, both off-line when the studied object is not in use, and on-line when the studied object is in use, allows to reduce the amount of material used to make objects. NDT is therefore not only an important component in quality control, but also in the larger endeavour to ensure sustainability.

Another strategy to make life more sustainable is the development of new materials. Researchers are actively working on new materials that rely less on rare minerals, are stronger, lighter, ecologically producible, easily recyclable, etc. than existing options. Before new materials can be used in industrial applications, their properties need to be known under as many circumstances as possible. To determine the properties of materials, once they are used in an application, it is often highly desirable to do so without destroying the considered object, as, e.g., prototyping is an expensive and time consuming practice. Non-destructive evaluation (NDE) is a desirable tool to obtain unknown material parameters.

Non-destructive testing and evaluation (NDT&E) is a large field of study which includes different disciplines. In the field of NDT&E researchers are developing and implementing non-destructive methods to acquire observations that are used to obtain answers to questions such as, "What are the properties of the material?", "Has the material been damaged?", "Where are the present defects?", "How long can the object be used before defects reach a critical point?". The most obvious non-destructive observation methods are touch and sight. When, e.g., checking a car for damage, you will generally look at it, and when you think you see some damage you will instinctively touch the suspicious patch to have a better sense of the extent of the damage. There are however accuracy limits to this universal method as too small defects will not be captured by our senses. Small defects may nevertheless have significant consequences. Therefore scientists keep investigating and developing more accurate methods to acquire and evaluate relevant observations.

The tools of touch and sight only allow to inspect the outer layers of an object, leaving internal defects well hidden. Investigating internal properties of an object non-destructively is more challenging than determining exterior properties. Using X-rays to inspect internal damage in living beings is a well known practice and a good example of NDT&E methods. It can be used to *look* inside a human body to assess, e.g., the extent of a bone fracture or the composition of tumours. A second example is the use of magnetic resonance imaging to study brain activity. Over the past decades, these techniques have become quite valuable in medical applications but are unfortunately limitedly applicable to study dense solid materials such as metals or new composite materials. Techniques to study internal properties of solids are less publicly known, but also exist and are not that different from the methods used in medicine. Internal inspection techniques generally rely on the use of mechanical waves, e.g., sound, and electromagnetic waves, e.g., light.

Most internal inspection techniques rely on extensive expertise to set-up an experiment and evaluate the results. Depending on the used measuring technique, the investigator will have to invest time and effort in consuming trial and error procedures. This can lead to unrealistic time requirements, e.g., to evaluate a batch of objects. Therefore, it is essential to investigate procedures for automated NDT&E purposes where the input of the investigator is reduced to acquiring a predefined set of measurements.

1.2 Problem statement

Every construction, apparatus and tool needs continuous monitoring or at least regular inspection in order to guarantee its correct functioning and safety. This quality control should be carried out in situ, in some cases even during operation. The test results should be accurate and reliable in order to allow the localisation of defects and to justify reparation. An early detection of damage is particularly advantageous because it simplifies logistics and avoids operational breakdown.

Many experimental and hybrid techniques, i.e., methods that combine measurements and simulations, for NDT&E exist, which all need significant input from an experienced operator. Yet, it is desirable to develop tools that can safely be operated by non-experts. There is also a need to allow the study of more complex set-ups and phenomena.

The existing techniques are often problem specific, leading to a large set-up and development cost. A more generic approach would reduce the cost of developing new procedures and specific set-ups.

Due to the complexity of the involved problems, hybrid methods are essential, but simulation can be demanding. Many numerical techniques to simulate wave propagation phenomena exist, hence selecting procedures that perform well enough to reach satisfactory results will take significant effort. We specifically need methods to efficiently solve wave propagation problems, keeping in mind that the computed results will be used for comparison with experimental data.

1.3 Research objectives

The main objective of this research is to develop software for automated NDT&E purposes based on wave propagation phenomena. The software should be more efficient and more widely applicable than the state-of-the-art, and should exclude the need of elaborate NDT&E expertise of the operator. It is our goal to develop software that can handle testing techniques harnessing acoustic, electromagnetic and elastic wave propagation. As these phenomena have shared properties, we aim to maintain and reinforce genericity of models and solution procedures throughout this work. The objective of this genericity is to allow an easier transfer of knowledge and experience between research fields using either one of the considered wave propagation phenomena.

1.4 Outline of the thesis

As wave phenomena are at the core of the investigated NDT&E techniques, we start with a description of the required theory to understand wave propagation problems in Chapter 2. The abstract wave equation will be introduced in one and multiple dimensions followed by an introduction of the acoustic, electromagnetic and elastic wave equation. In Chapter 2 we will define forward and inverse wave propagation problems, followed by an introduction on how these will be solved.

In Chapter 3 we study the finite integration technique to simulate forward wave propagation problems as introduced in Chapter 2. Several time stepping methods will be introduced to solve the spatially discretised problem in time. The properties of the different time integrators will be studied in multiple examples. The main focus of the chapter is to introduce a framework of staggered grids, which allows to present a spatial discretisation of the acoustic, electromagnetic and elastic wave problems in a unified manner.

In Chapter 4 we use the finite element method for the spatial discretisation of the wave equations from Chapter 2. We consider the use of continuous conforming elements and discontinuous elements. For the discretisation of the elastic wave equation, special attention has to be given to the symmetry of the discretised stress tensor. We impose the symmetry differently depending on the used elements. This motivates a thorough comparison to determine which method is most suitable to use in inverse NDT&E problems.

We focus on solving inverse wave problems in Chapter 5. An overview of typical inverse problems in NDT&E is given, followed by a rigorous problem formulation. We propose to solve the inverse problems of interest as optimisation problems constrained by wave equations with the use of gradient information. The difficulties involved in solving these problems will be explained and treated. We will illustrate the proposed procedure to retrieve unknown wave speeds and compare with current day gradient-free methods.

Absorbing layers for domain truncation are introduced in Chapter 6. Truncation of domains is required in the case of infinite domains, but can also be useful to trim irrelevant parts of a large domain to limit the computational domain to the domain of interest. One difficulty in implementing absorbing layers is to set the involved parameters. In Chapter 6 we propose an automated calibration technique for absorbing layers. An optimisation problem is solved at the core of the calibration procedure. To solve this optimisation problem, we use the methods that have also been used to solve inverse problems in Chapter 5.

We conclude the thesis in Chapter 7 with a summary, followed by conclusions of the conducted research and suggestions for future work.

Chapter 2

Preliminaries

In this chapter we define the motion of waves and their main properties by introducing wave functions. We introduce mathematical models called wave equations to describe how waves propagate. The first two models are abstract in the sense that the involved quantities are not defined as physical phenomena. Models for acoustic, electromagnetic and elastic wave propagation are also introduced. Next, we define forward and inverse wave propagation problems and give an outlook on how to solve them.

2.1 Waves

A wave is an oscillation that transfers energy from one point to another through space or mass over a certain amount of time. Hence a wave is a space and time dependent phenomenon. There are two main types of waves: mechanical and electromagnetic waves. Mechanical waves propagate through a medium, while electromagnetic waves do not require a medium to propagate. Mechanical waves can be subdivided further in elastic bulk and surface waves. We refer to elastic waves propagating through media that do not support shear stresses as acoustic waves. Electromagnetic waves consist of periodic oscillations of electrical and magnetic fields generated by charged particles, and can travel through vacuum.

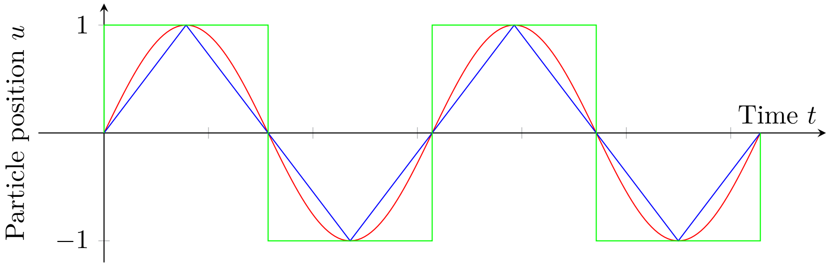


Figure 2.1: Height u of three different particles oscillating in space, as a function of time.

2.1.1 Wave function

The physical quantities involved in wave propagation can be described by mathematical expressions which we call *wave functions*. For instance, consider the height u of a single particle moving up and down in space. The graph shown in Figure 2.1 depicts the height of three independent particles as a function of time. The initial position at time $t = 0$ of all particles is $u = 0$, and all particles oscillate between $u = -1$ and $u = 1$. The green line describes the height of a particle that instantly jumps from one extreme position to another. The blue lines describes the height of a particle of which the direction of the velocity changes instantly at extreme positions. The red line describes the trajectory of a particle oscillating in a smooth manner, i.e., without large instantaneous changes in position, velocity or acceleration. All these motions can be mathematically described by wave functions. However, only smooth kind of waves will be studied here. For instance, the smooth line in Figure 2.1 can be described by the function $u = \sin(t)$.

2.1.2 Wave properties

A first wave property is the period T, which is the time the particle needs to move up and down once, starting from its lowest position. The frequency of the wave f is the number of times a particle goes up and down during one unit of time. The period of a wave is the inverse of the frequency $T = 1/f$. Next, we refer to half the distance between the highest and lowest position of the particle as the amplitude of the wave A. Figure 2.2 illustrates these properties. The red and blue trajectory have a higher amplitude than the black trajectory, while the blue and red trajectory have a lower frequency, and hence longer period than the blue trajectory.

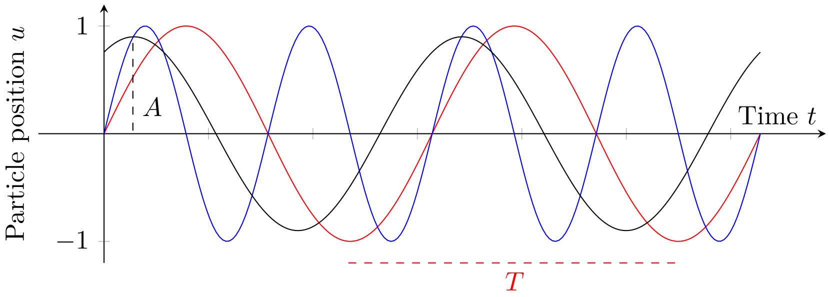


Figure 2.2: Height of independent particles moving up and down in space, as a function of time. The blue line shows the height of a particle oscillating twice as fast as the particle indicated by the red line. The black line shows the height of a particle oscillating with a smaller amplitude A, indicated by a vertical dashed black line, than the particle indicated by the red line. The period T of the red oscillation is indicated by a horizontal dashed red line.

So far, we have considered the time-dependent properties of a single particle moving up and down in space. To study the space dependent properties of a wave, consider the left-most particle of a slinky[1], to move, causing the neighbouring particle, as it is connected to the first particle, to move also. As all particles of the slinky are connected to one another, the movement of the left-most particle will propagate throughout the entire slinky. At the top of Figure 2.3 a slinky is depicted where the left-most particle oscillates up and down as described before, resulting in a transversal wave propagating in the slinky. Transversal waves oscillate in a direction perpendicular to the direction in which the wave travels. At the bottom of Figure 2.3 a slinky is depicted where the left-most particle oscillates from left to right, resulting in a longitudinal wave propagating in the slinky. Longitudinal waves oscillate in the same direction as the wave travels.

The positions with the lowest amplitudes of a transversal wave are called troughs. The places in a longitudinal wave with highest density of particles are called compressions. The distance between two consecutive troughs or compressions is called the wavelength λ. The behaviour of the left-most particle is transferred from one particle to another, from left to right. We refer to the speed with which this behaviour moves as the wave speed c. The wave speed depends on the material out of which the medium is made and relates the frequency of the left-most particle to the wavelength through the relation $c = \lambda f$. Longitudinal waves move about twice as fast as transversal waves [65]. We denote the longitudinal and transversal wave speeds as c_l and c_t respectively.

---

[1] A slinky is a precompressed helical spring.

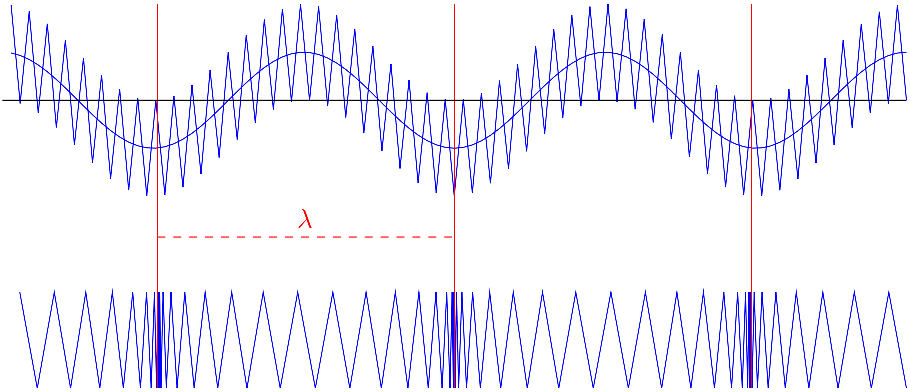


Figure 2.3: A transversal wave propagating through a slinky (top). A longitudinal wave propagating through a slinky (bottom). The troughs of the transversal wave and compressions of the longitudinal waves are indicated by solid red lines. The wavelength λ is indicated by a dashed line.

The above described waves are called harmonic waves, and are mathematically described as

$$u(x,t) = A\sin(kx \pm \omega t), \tag{2.1}$$

where $k = 2\pi/\lambda$ is the wave number and $\omega = 2\pi f$ is the angular wave frequency. Other wave functions exist, and are described as a superposition of sine and cosine functions. In general, a wave is described by a mathematical function of the form

$$f\left(kx \pm \omega t\right), \tag{2.2}$$

where $f(kx)$ describes the initial situation.

2.2 Wave equations

Wave equations describe how oscillating particles interact with their surroundings. Wave equations are a particular kind of differential equation, namely hyperbolic partial differential equations. Partial differential equations (PDEs) relate the functions that are used to describe the involved physical magnitudes with their derivatives, which indicate the rate of change, often with respect to space and time.

Wave equations set out how a disturbance proceeds over time and through space. The mathematical form of this equation and the spatial differential operators vary depending on the type of wave. In this section, we introduce the wave equation for an abstract wave in one dimension and three dimensions. In the subsequent subsections, the acoustic, electromagnetic and elastic wave equations are introduced and shaped into a generic form that will be suggested while introducing the abstract wave equation in three dimensions.

2.2.1 Abstract wave equation in one dimension

The abstract wave equation in one dimension is

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}, \tag{2.3}$$

where the function $u(x,t)$ represents a physical quantity depending on one spatial coordinate x and time t. Wave propagation is the result of interaction between two physical quantities, which motivates to express the wave equation as a system of first order equations. We introduce the function $v(x,t)$ to describe the second involved physical quantity, apart from $u(x,t)$, and the scalars a and b representing material properties[2], such that $c^2 = ab$, which allows us to formulate the wave equation as the mixed wave system

$$\tfrac{\partial u(x,t)}{\partial t} = a \tfrac{\partial v(x,t)}{\partial x}, \tag{2.4}$$

$$\tfrac{\partial v(x,t)}{\partial t} = b \tfrac{\partial u(x,t)}{\partial x}. \tag{2.5}$$

2.2.2 Abstract wave equation in three dimensions

The abstract wave equation in three dimensions is

$$\begin{aligned} \tfrac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t} &= \boldsymbol{a}\mathrm{D}^v_{\boldsymbol{x}}\boldsymbol{v}\left(\boldsymbol{x},t\right), \\ \tfrac{\partial \boldsymbol{v}(\boldsymbol{x},t)}{\partial t} &= \boldsymbol{b}\mathrm{D}^u_{\boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right) + \boldsymbol{f}\left(\boldsymbol{x},t\right), \end{aligned} \tag{2.6}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are tensor fields of zeroth, first or second order[3][4] used to describe the involved physical quantities, $\boldsymbol{x}$ is a vector containing the spatial coordinates, $\boldsymbol{f}$ is a tensorial source function, $\boldsymbol{a}$ and $\boldsymbol{b}$ are tensors containing the material properties of the considered medium and $\mathrm{D}^v_{\boldsymbol{x}}$ and $\mathrm{D}^u_{\boldsymbol{x}}$ are spatial differential

---

[2] We will consider spatially varying material parameters in the physical wave problems later on. To keep the notation for the abstract formulations as clear as possible, we assume constant material parameters.

[3] A zeroth order tensor field is also known as a scalar field and a first order tensor field is also known as a vectorial field.

[4] The tensorial order is not necessarily the same for both involved physical quantities.

operators, such as a divergence, gradient or curl, depending on the tensorial order of the involved physical quantities.

We reshape (2.6) into the generic form

$$\dot{\boldsymbol{q}} + \sum_{i=1}^{3} \boldsymbol{F}_{i,i} = \boldsymbol{f}, \tag{2.7}$$

where $\boldsymbol{q}$ is a vector of length N_q containing all components of the unknown tensor fields, $\boldsymbol{F}_i$ are flux vectors of length N_q and $\boldsymbol{f}$ is a source function of length N_q. In the case of linear wave propagation, the flux vectors can be described as $\boldsymbol{F}_i = \mathbf{A}_i \boldsymbol{q}$, where the matrices $\mathbf{A}_i$ contain material parameters and are defined for each specific problem together with problem specific definitions for the vector $\boldsymbol{q}$ in the following sections. The notation $\boldsymbol{F}_{i,i} = \partial \boldsymbol{F}_i / \partial x_i$ (no summation) implies component-wise partial differentiation of $\boldsymbol{F}_i$ with respect to x_i. The dot above a symbol indicates a time derivative. To keep the text as readable as possible, the spatial and temporal dependencies are dropped as much as possible.

2.2.3 Acoustic wave equation

Acoustic waves are longitudinal mechanical waves travelling at the speed of sound, which is determined by the medium in which the waves travel.

Acoustic wave propagation can be described by the system

$$\begin{aligned} \frac{1}{K}\dot{p} &= -\nabla \cdot \boldsymbol{v}, \\ \rho \dot{\boldsymbol{v}} &= -\nabla p + \boldsymbol{f}, \end{aligned} \tag{2.8}$$

where $K > 0$ is the bulk modulus, p is the pressure, $\boldsymbol{v} = (v_1, v_2, v_3)$ is the particle velocity, $\rho > 0$ is the mass density, and $\boldsymbol{f}$ is an applied body force. This problem is transformed into the generic form (2.7) by choosing $\boldsymbol{q} = (v_1, v_2, v_3, p)^T$. The matrices $\mathbf{A}_i$ for the acoustic wave problem are

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 0 & \rho^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K & 0 & 0 & 0 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & K & 0 & 0 \end{pmatrix}, \mathbf{A}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho^{-1} \\ 0 & 0 & K & 0 \end{pmatrix}. \tag{2.9}$$

2.2.4 Electromagnetic wave equation

In contrast to acoustic waves, electromagnetic waves do not require a medium to travel in. Electromagnetic waves are transversal waves that travel at the speed of light, which is constant in vacuum, or determined by the medium in which the waves travel.

Electromagnetic wave propagation can be described by the system

$$\begin{aligned} \mu \dot{\boldsymbol{H}} &= -\nabla \times \boldsymbol{E}, \\ \varepsilon \dot{\boldsymbol{E}} &= \nabla \times \boldsymbol{H} - \boldsymbol{J}, \end{aligned} \tag{2.10}$$

where $\mu > 0$ is the permeability, $\boldsymbol{H} = (H_1, H_2, H_3)$ is the magnetic field strength, $\boldsymbol{E} = (E_1, E_2, E_3)$ is the electric field strength, $\varepsilon > 0$ is the permittivity, and $\boldsymbol{J}$ is a current density. This problem is transformed into the generic form (2.7) by choosing $\boldsymbol{q} = (H_1, H_2, H_3, E_1, E_2, E_3)^T$. The matrices $\mathbf{A}_i$ for the electromagnetic wave problem are

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu^{-1} \\ 0 & 0 & 0 & 0 & \mu^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon^{-1} & 0 & 0 & 0 \\ 0 & -\varepsilon^{-1} & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \mu^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mu^{-1} & 0 & 0 \\ 0 & 0 & -\varepsilon^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon^{-1} & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & -\mu^{-1} & 0 \\ 0 & 0 & 0 & \mu^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon^{-1} & 0 & 0 & 0 & 0 \\ -\varepsilon^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.11}$$

2.2.5 Elastic wave equation

Elastic waves are mechanical waves that can travel both transversally or longitudinally in solids. The longitudinal wave speed is about twice the transversal wave speed [65] and is determined by the medium in which the waves propagate.

Elastic wave propagation can be described by the system

$$\begin{aligned} \rho\dot{\boldsymbol{v}} &= \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f}, \\ \boldsymbol{\mathcal{C}}^{\mathbf{-1}} : \dot{\boldsymbol{\sigma}} &= \frac{1}{2}\left(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T\right), \end{aligned} \tag{2.12}$$

where $\boldsymbol{\mathcal{C}}$ is the fourth-order elastic stiffness tensor, $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{v} = (v_1, v_2, v_3)$ is the particle velocity, $\rho > 0$ the mass density, and $\boldsymbol{f}$ is an applied body force. This problem is transformed into the generic form (2.7) by choosing $\boldsymbol{q} = (v_1, v_2, v_3, \epsilon_{11}, \epsilon_{22}, \epsilon_{33}, 2\epsilon_{23}, 2\epsilon_{13}, 2\epsilon_{12})^T$, where

$$\boldsymbol{\epsilon} = \boldsymbol{\mathcal{C}}^{\mathbf{-1}} : \boldsymbol{\sigma} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{12} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{13} & \epsilon_{23} & \epsilon_{33} \end{pmatrix} \tag{2.13}$$

is the strain tensor.

The matrices $\mathbf{A}_i$ for the elastic wave problem are

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 0 & -\frac{C_{11}}{\rho} & -\frac{C_{12}}{\rho} & \frac{C_{13}}{\rho} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{C_{66}}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{C_{55}}{\rho} & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{C_{66}}{\rho} \\ 0 & 0 & 0 & -\frac{C_{21}}{\rho} & -\frac{C_{22}}{\rho} & -\frac{C_{23}}{\rho} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{c_{44}}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{C_{55}}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{C_{44}}{\rho} & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_{31}}{\rho} & -\frac{C_{32}}{\rho} & -\frac{C_{33}}{\rho} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad (2.14)$$

where the variables C_{ij} are elements from the fourth-order elastic stiffness tensor $\boldsymbol{\mathcal{C}}$ such that

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} = \underbrace{\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{pmatrix}, \qquad (2.15)$$

where $\mathbf{C}$ is called the *stiffness matrix*.

2.3 Wave propagation problems

In this thesis we study forward and inverse wave propagation problems. Besides wave equations, more information is needed to define a wave problem. We start by formulating forward wave problems, highlighting the needed additional information. Next, inverse wave problems are formulated.

2.3.1 Forward wave propagation problems

A forward wave problem reads: solve the wave equation (2.7) to find $\boldsymbol{q}\left(\boldsymbol{x},t\right)$ for all $\boldsymbol{x}$ in a domain of interest Ω and all t in a time interval $[0,T]$ with final time T, given specific material parameters, the source function $\boldsymbol{f}$, the initial condition $\boldsymbol{q}\left(\boldsymbol{x},0\right)=\boldsymbol{q}_0$ and the boundary behaviour, which will be discussed in the following paragraph. We refer to the involved wave equation as *the model* $\mathbf{M}$ and denote the collection of all given parameters as $\mathbf{P}$. We denote a wave problem in short as $\mathbf{M}\left(\mathbf{P}\right)=\boldsymbol{q}\left(\boldsymbol{x},t\right)$.

Boundary conditions

When solving a wave equation, we are interested in the solution on the domain and time interval of interest. Larger domains of interest or longer time intervals will often complicate the solution and computations. It is therefore undesirable to compute solutions in infinitely large domains or for infinitely long time intervals. This implies the use of finite time intervals and finite domains with boundaries or the truncation of infinite domains using open boundaries. Defining the behaviour of the boundaries of the domain is essential to completely formulate the forward problem.

Regardless of the physical problem, several common boundary interactions exist. The first is the reflection of the wave. As there are two unknown fields, either one can be reflected, leading to two different ways of reflecting the wave. In practice, the energy of the wave will partially be absorbed by the reflecting boundary. We however always assume perfect reflection. Open boundaries are used to mimic infinite domains and can be realised by absorbing layers in front of reflecting boundaries. Handling this kind of boundary condition is more difficult and will be discussed in Chapter 6. A boundary condition can be used to excite the system, by inserting energy in the domain. All above mentioned boundary conditions can be implemented by imposing a *Dirichlet boundary condition*, i.e., imposing the solution on one of the involved unknown fields. In addition, *Neumann boundary conditions* exist, which impose the derivative on one of the involved unknown fields. We will only use homogeneous Neumann boundary conditions.

Energy

When studying wave propagation problems, it is useful to monitor the energy in the system. The energy conservation law formulated in the Poynting theorem [58, 70] states for all wave problems that in absence of attenuation the change in time of the stored energy should equal the energy per unit time entering the model through the boundaries or via the source field. Hence computing the energy in the system of a wave problem is a useful tool to evaluate solutions of wave problems.

The energy in the system of an acoustic wave problem is

$$E_{\text{acoustic}}(t) = \frac{1}{2}\int_{\Omega} \rho \boldsymbol{v} \cdot \boldsymbol{v} + \frac{1}{K} p^2 \,\mathrm{d}x. \tag{2.16}$$

The energy in the system of an electromagnetic wave problem is

$$E_{\text{electromagnetic}}(t) = \frac{1}{2}\int_{\Omega} \mu \boldsymbol{H} \cdot \boldsymbol{H} + \varepsilon \boldsymbol{E} \cdot \boldsymbol{E} \,\mathrm{d}x. \tag{2.17}$$

The energy in the system of an elastic wave problem is

$$E_{\text{elastic}}(t) = \frac{1}{2}\int_{\Omega} \rho \boldsymbol{v} \cdot \boldsymbol{v} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon} \,\mathrm{d}x. \tag{2.18}$$

2.3.2 Inverse wave propagation problems

An inverse wave propagation problems reads: given a model $\mathbf{M}$ and observed output data $\boldsymbol{q}_{\text{obs}}$, retrieve the unknown value of one or more parameters from the parameter set $\mathbf{P}$ as defined in Section 2.3.1 such that the computed solution $\mathbf{M}(\mathbf{P})$ matches the observed data $\boldsymbol{q}_{\text{obs}}$. Ideally we find exactly one collection of parameters that results in an exact equality.

2.4 Solving wave propagation problems

In this thesis, we use numerical techniques to solve both forward and inverse wave problems. Our computations will result in approximations to the exact, unknown solution. We will refer to the obtained approximations as solutions to the wave problem. When relevant, we will discuss the accuracy of the obtained approximations.

2.4.1 Solving forward wave propagation problems

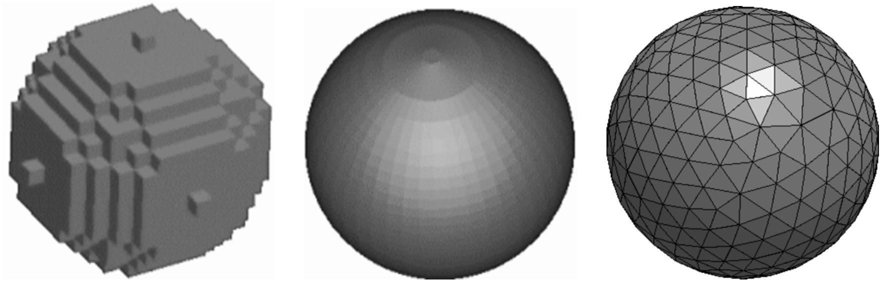
Many techniques exist to numerically solve wave problems. Some are general techniques to solve PDEs, others are more specific for wave problems. In this thesis, we will introduce the most common techniques to solve PDEs.

The results of numerically solving a continuous wave problem as described in Section 2.3.1 are the exact solutions of a discrete wave problem, derived from a continuous wave problem. We derive the discrete wave problem by first discretising the spatial dimensions, followed by discretising time.

Discrete approximations in space

To use spatial discretisation methods, the computational domain $\Omega \subset \mathbb{R}^d$ has to be subdivided in cells. For instance, Figure 2.4 shows three different meshes for a sphere. Figure 2.4a shows a staircase mesh consisting of cubical cells. Figure 2.4b is a structured conforming mesh consisting of non-rectangular cells. A conforming mesh approximates the computational domain better than a staircase mesh and can be obtained by using a transformation on a rectangular mesh. The needed transformations are problem specific and are rarely available for arbitrary geometries. Figure 2.4c shows an unstructured grid using tetrahedral cells. In unstructured grids, cells can each have a different size and shape, but are commonly formed by triangles or tetrahedra as these shapes allow arbitrary good approximation of any realistic geometry.

Regardless of the type of mesh, a certain number of discretisation points is used by the spatial discretisation method (see following section) to approximate the involved wave functions. Figure 2.5 shows two wavelengths of a wave (red) and approximations that rely on a different number of discretisation points (samples) of the wave. The green line uses five discrete points, which results in the approximation $u = 0$, which does not allow to retrieve the amplitude or frequency of the wave. The blue line, on the other hand, with ten samples per period, allows to extract fair approximations of the wave properties. The samples in Figure 2.5 are connected by straight lines, which corresponds to first order elements or linear interpolation. It is however also possible to use higher order approximation techniques, in which case multiple discretisation points are working together to form higher order approximation functions. More



(a) Staircase mesh. (b) Structured conforming mesh. (c) Unstructured tetrahedral mesh.

Figure 2.4: Meshing a sphere

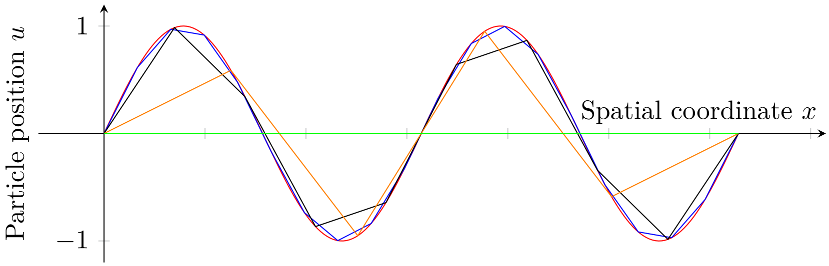


Figure 2.5: Discrete approximation of two wavelengths of a wave (red) with 20 (blue), 10 (black), 6 (orange) and 5 equidistant discretisation points.

samples and higher order approximation functions lead to a higher accuracy of the approximation, but also to a higher computational cost. To keep the computational cost low, as few discretisation points as possible should be used to obtain a desired accuracy. However, when too few discretisation points are used, numerical errors will be induced, resulting in a useless solution. When using lowest order approximations, it is recommended to use about twenty discretisation points per wavelength to suppress numerical dispersion, i.e., waves travelling at different wave speeds due to numerical errors. The small sharp peaks in the red curve on the interval $[0.5, 1]$ in Figure 2.6 illustrate numerical dispersion [72]. Note that there are often multiple wave frequencies and hence multiple wavelengths in play in one simulation. In that case, we generally want to meet the requirements for the smallest relevant wavelength in the problem.

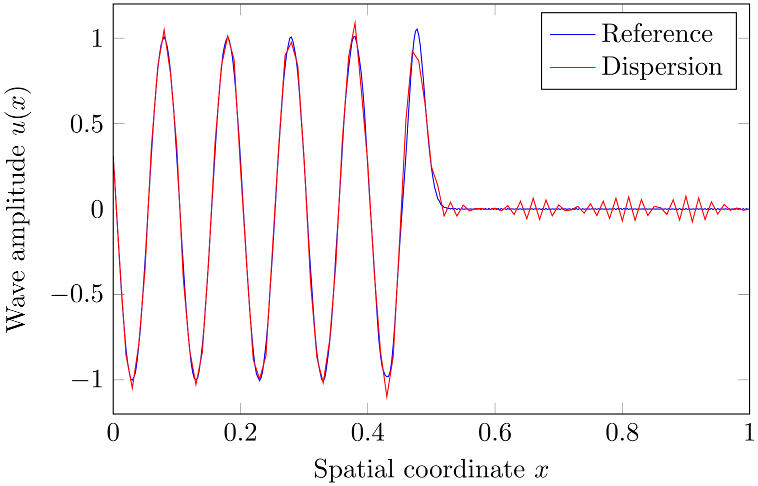


Figure 2.6: Numerically computed amplitude of a wave. The blue curve was simulated with sufficient cells and time steps to avoid numerical dispersion. The red curve was simulated with only ten cells per wavelength, resulting in small sharp peaks on the interval $[0.5, 1]$ which illustrates numerical dispersion.

Spatial discretisation methods

A spatial discretisation method is used to approximate the continuous problem on the chosen mesh. Staircase and transformed structured meshes allow spatial discretisation by the finite difference method (FDM) or finite integration technique (FIT). Standard FIT and FDM are fast methods with low complexity. However, the required use of structured meshes is often prohibiting to approximate complex geometries. For simple geometries, FIT and FDM are the preferred methods.

The finite volume method (FVM) is a technique with comparable complexity to FDM/FIT that can be used on unstructured meshes. Standard FVM does however not allow higher order accuracy, which is desirable as will be shown in Chapter 4. Alternatively, finite element methods (FEM) can also handle unstructured grids and do allow higher order accuracy, but have problems fulfilling conservation laws when discontinuities occur. FEM has a higher complexity than FIT/FDM and FVM. To combine the best of FEM and FVM, researchers have suggested the Discontinuous-Galerkin FEM (DG-FEM). DG-FEM is more complex than FEM, but fulfils conservation laws in all cases and can be used for higher order accurate computations. This discussion is summarised in Table 2.1 [51]. The use of FIT/FDM for solving forward wave

	FIT/FDM	FVM	FEM	DG-FEM
Complex geometries	×	✓	✓	✓
Higher-order accuracy	✓	×	✓	✓
Conservation laws	✓	✓	(✓)	✓
Method complexity	1	1	2	3

Table 2.1: Comparison between several common spatial discretisation techniques.

propagation problems will be introduced in Chapter 3. The application of (DG-)FEM for solving forward wave propagation problems will be discussed in Chapter 4. FVM will not be considered separately, as it can be formulated as a lowest order DG-FEM.

Discrete approximations in time and time integrators

After meshing and spatial discretisation, a time integrator is used to update a solution $\boldsymbol{q}_n$ at a current time t_n to the solution $\boldsymbol{q}_{n+1}$ at a following time instant $t_{n+1} = t_n + \Delta t$. If the time step size Δt is too large, the solution will be inaccurate. A minimum of twenty time steps per period is suggested to suppress numerical dissipation [85], i.e., loss of energy in the system due to approximation errors. The lower peak amplitudes of the red curve in Figure 2.7 for larger values of x illustrates numerical dissipation.

Time integrators can be subdivided into implicit and explicit methods. An explicit time integrator f only uses available information from previous time steps when computing the solution $\boldsymbol{q}_{n+1} = f\left(\boldsymbol{q}_n, \boldsymbol{q}_{n-1}, \cdots \boldsymbol{q}_0\right)$ from one time step to another. An implicit method g on the other hand also depends on the new solution, i.e., $\boldsymbol{q}_{n+1} = g\left(\boldsymbol{q}_{n+1}, \boldsymbol{q}_n, \cdots \boldsymbol{q}_0\right)$. One step with an implicit time integrator typically takes more computational effort than one step with an explicit method.

Specific time integration methods can also impose additional requirements to the time step size for the solution to remain stable. Within one time step, a wave can travel a distance of $c\Delta t$ in any direction. Hence, the domain in which a wave can propagate is restricted by the space that is within a range of $c\Delta t$ in every direction. We refer to this domain as the continuous domain. An intuitive requirement for stability is that the continuous domain has to be enclosed by the discrete domain, which is the convex hull of the used discretisation points. This requirement is called the Courant-Friedrichs-Lewy (CFL) condition [31]. As the discrete domain for implicit methods equals the entire computational domain, the CFL condition will not be a problem. For explicit methods however, only adjacent discretisation points are used, meaning that the CFL condition will restrict the time step size. For a lowest order method in one dimension, the CFL condition reads

$$\Delta t \leq \frac{\Delta x}{c}, \tag{2.19}$$

where Δx is the space between two spatial discretisation points and c is the largest occurring wave speed. If the spatial discretisation uses N_p discretisation points per wavelength, i.e, $N_p = \lambda/\Delta x$, then

$$\Delta t \leq \frac{\Delta x}{c} \quad \Longleftrightarrow \quad \Delta t \leq \frac{\lambda}{N_p c} \quad \Longleftrightarrow \quad \Delta t \leq \frac{T}{N_p}, \tag{2.20}$$

where λ is the wavelength and T is the wave period. Hence, the temporal resolution should be at least as high as the spatial resolution, which is usually a fair requirement.

In multiple dimensions, the CFL condition is harder to write down. For unstructured grids there is no strict relation available. For a lowest order method in d dimensions using a staircase grid with uniform element size Δx [73], the CFL condition reads

$$\Delta t \leq \frac{\Delta x}{c\sqrt{d}} \quad \Longleftrightarrow \quad \Delta t \leq \frac{T}{N_p\sqrt{d}}, \tag{2.21}$$

meaning that the temporal resolution has to be at least a factor $\sqrt{d}$ larger than the spatial resolution, increasing the required number of time steps. Since implicit methods do not have this restriction, they require fewer time steps in multiple dimensions.

Accuracy of the numerical approximation

The accuracy of the computed solution of a forward problem depends on the used mesh, spatial discretisation method and time integrator. Each mesh and spatial discretisation method induce a *spatial discretisation error* which is independent of the selected time integrator. The accuracy of the computed solution will be determined by the spatial discretisation error when a time integrator is used that is more accurate than the spatial discretisation. Each time integrator induces a *temporal discretisation error* which is independent of the selected spatial discretisation. The accuracy of the computed solution will be determined by the temporal discretisation error when a mesh and spatial discretisation method is used that are more accurate than the time integrator.

In practice, it is preferable to have a spatial and temporal discretisation error of about the same magnitude. If one is significantly larger than the other, then too much computational work has to be done. Either the spatial discretisation or

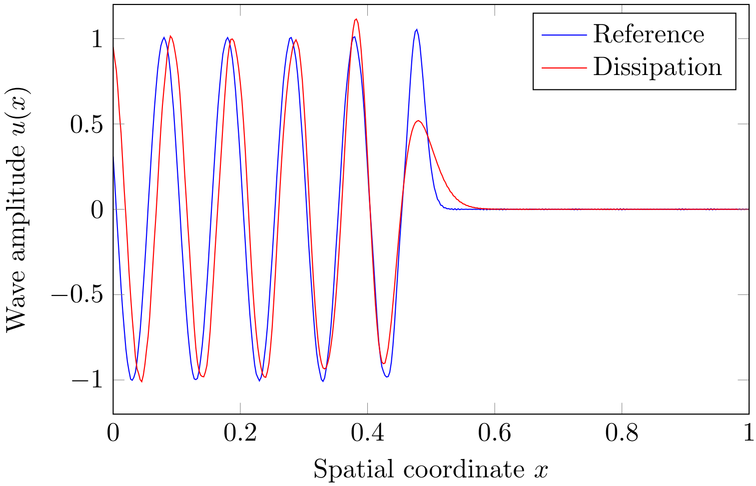


Figure 2.7: Numerically computed amplitude of a wave. The blue curve was simulated with sufficient cells and time steps to avoid numerical dissipation. The red curve was simulated with only ten time steps per period, resulting in lower peak amplitudes compared to the reference which indicates numerical dissipation.

the time integration could have been realised with relaxed parameters, leading to faster computation for an equally accurate approximation. In the following chapters, we will test multiple combinations of spatial discretisations and time integrators to avoid computations that do not lead to a higher accuracy.

2.4.2 Solving inverse wave propagation problems

Solving inverse wave problems can be performed using several techniques. Sometimes, it is possible to invert the model $\mathbf{M}$ to obtain a new model $\mathbf{M}^{-1}$, which allows formulating the inverse problem as a forward problem. We however consider inverse problems that start from an input consisting of measured reference data, which can only be modelled using stochastic models due to noise and measurement errors. Hence, an inverse model does not exist for our wave problems.

As there is no inverse model available, we can only evaluate the forward model with chosen values for the unknown parameters. The main challenge is to use as few guesses as possible to retrieve a likely candidate for the unknown parameters. We will consider methods that only use the forward model, as well as methods that use gradient information of the forward model. We focus on solving inverse problems in NDT&E in Chapter 5.

2.5 Software for solving wave propagation problems

As the main aim of this work is to develop efficient software tools, we give an overview of which software tools that will be used in this work.

To study the finite integration method in Chapter 3, an implementation was made in Matlab [71]. The reason for writing new code for this method is that we wanted to make sure that the unified approach that will be presented is implemented consistently, allowing an easy comparison between nodal and mimetic allocation (see Section 3.2.2). We also need a toolbox that treats simulations for acoustic, electromagnetic and elastic problems in a similar way, in order to make sure that it is fair to compare the obtained timings. Matlab is a good environment to test out new implementations, but does not result in the most efficient computer code. Hence it would not be fair to compare the results from chapter 3 with other mentioned results.

In the remaining chapters, the finite element method was used. We needed a toolbox which provides a tool-chain with the required options to solve our problem, such as, e.g, specific conforming continuous element spaces as will be described in Section 4.3. Using commercial software such as, e.g., COMSOL Multiphysics [1] was excluded, as this kind of software does not allow sufficient access to underlying routines to model, e.g., non-standard perfectly matched layer as will be studied in Chapter 6. Several open source libraries offer the needed features for solving wave propagation problems, such as, a.o., FEniCS/DOLFIN [6, 67, 68] and Deal.II [2]. We chose to use the FEniCS/DOLFIN library mainly because of the compatibility with dolfin-adjoint [38, 44], which can derive the gradient information that will be used in Chapters 5 and 6 automatically from forward solvers implemented using the FEniCS/DOLFIN library. FEniCS/DOLFIN is mainly written in C++ but also includes Python code. In this work, the Python interface to FEniCS/DOLFIN was used, as this is a requirement to use the dolfin-adjoint library. Dolfin-adjoint is written purely in Python, but uses the libadjoint library [36] which is implemented in C++.

Our choice to use FEniCS/DOLFIN and dolfin-adjoint resulted in the fact that minimal work was required to achieve working code for the problems we set out to tackle. Only two additions were made to these toolboxes for our studies. The first addition is a C++ implementation in DOLFIN of an efficient linear solver that exploits the structure of the linear system resulting from using discontinuous elements. The second addition is a Python implementation in dolfin-adjoint of an objective functional that uses point-wise evaluations of the computed vector fields (see Section 5.5.7). Besides these additions, most implementation efforts were done to write FEniCS/DOLFIN scripts using the Python interface. We will include some FEniCS/DOLFIN code in Chapter 4 to demonstrate why FEniCS/DOLFIN is a good choice to handle the problems we will study. The use of dolfin-adjoint is illustrated in a code snippet in Chapter 5.

Chapter 3

Solving forward wave propagation problems using the finite integration technique

In this chapter, we use the finite integration technique (FIT) to solve wave equations. This technique is equipped with a concise notation [99], is related to differential form theory [20], and has been implemented as a mimetic discretisation for the electromagnetic case [28]. Mimetic methods organise the spatial discretisation to mimic the continuous problem as well as possible such that the canonical properties of the continuous problem are transferred to the discrete problem [55, 57].

The equivalence between FIT and the finite difference method (FDM) is described for the electromagnetic wave equation in, a.o., [20, 99]. We highlight the resemblance between FIT and FDM also for acoustic and elastic wave problems, and show that both methods result in a mimetic discretisation. The main goal of the chapter is to provide tools and concepts that allow a unified approach to apply FIT and FDM to wave propagation problems. For this, a concise notation for staggered grids will be introduced. We also show that all three considered wave problems fit the same semi-discrete formulation using either FIT or FDM. This is advantageous for designing multi-physics solvers.

We apply FIT to the acoustic, electromagnetic or elastic wave equation as introduced in Chapter 2 and refer to the corresponding methods as AFIT [70], EMFIT [98] and EFIT [39] respectively. Several time integrators are considered to solve the obtained spatially discretised problems. We compare the use of higher order time integration methods to the classical leapfrog method in order to determine the most suitable integrator for our purposes.

Despite the fact that the discussion in this chapter is mainly done for three dimensional problems, it is also valid for one and two dimensional problems. In different dimensions, different mesh elements, i.e., nodes, edges, faces and volumes, play different roles. We use the term *cell* for mesh elements of the same spatial dimension as the computational domain. A cell is a volume, face or edge in 3D, 2D and 1D respectively. Analogously, we use the term *facet* for mesh elements of one spatial dimension less than the computational domain. A facet is a face, edge or node in 3D, 2D and 1D respectively.

3.1 Unified framework for staggered grids

Historically, the electromagnetic wave equation (2.10) was the first of the three considered problems to be solved with FIT. From the introduction of FIT, staggered grids were used to allocate degrees of freedom (dofs) on edges and faces of the mesh [98]. This allocation is used to guarantee that the continuity characteristics of the discrete fields mimic those of the continuous fields. E.g., the electric field $\boldsymbol{E}$ has tangential continuity on material interfaces, which can be realised by allocating the electric field dofs on edges. As a second and more general example, we note that the normal components of fields representing fluxes are continuous, which is realised by allocating the dofs representing these fields on facets. In a later stage FIT was also applied to the other wave equations (2.8) and (2.12) [16, 70]. Here a different strategy was employed: staggered grids were used to allocate dofs on shifted nodes. This resulted in the same spatial discretisation as with finite difference methods. Both approaches, allocation of dofs on edges, faces, and volumes versus allocation on shifted nodes, are valuable as will be argued in the current and following sections.

In this section, we generalise the above two staggered grid approaches, and show that they can both be used on all three wave problems. For this, we construct eight staggered grids, starting from a structured mesh. We use tensor-product grids for which there is a one-to-one relation between the nodal coordinates and a *grid index* (i_1, i_2, i_3), with $i_j = 1, \ldots, I_j$ and I_j the number of nodes in j-direction ($j = 1, 2, 3$) or a *canonical index* $i = 1 \ldots N := I_1 I_2 I_3$, which runs over all nodes in the mesh. We restrict ourselves to equidistant Cartesian grids, which allow a more intuitive introduction of the concise notation. Generalisation to non-equidistant grids is in principle straightforward, as shown in [16].

3.1.1 Concise notation for staggered grids

Consider the primary grid $\mathcal{G}$, consisting of primary nodes $\mathbf{n}_i$, primary edges $\mathbf{e}_i^j$, primary faces $\mathbf{f}_i^j$ and primary volumes $\mathbf{v}_i$ (Fig. 3.1a). The upper index j indicates the direction of the edges and faces, i.e., the axis parallel to an edge or the axis parallel to the normal vector on a face. Seven additional grids, staggered with respect to each other, are constructed by shifting the primary grid by half a cell width in one, two or three directions. The *uni-axially shifted grids* and their corresponding nodes, edges, faces and volumes are denoted by $\left(\widetilde{\mathcal{G}}^j, \widetilde{\mathbf{n}}_i^j, \widetilde{\mathbf{e}}_i^j, \widetilde{\mathbf{f}}_i^j, \widetilde{\mathbf{v}}_i^j\right)$ where the upper index j indicates the direction along which the grid is shifted (Fig. 3.1b). Note the double meaning of the upper index. For example $\mathbf{e}_i^1$ is the edge emanating from node $\mathbf{n}_i$ and running in the positive x_1-direction. Edge $\widetilde{\mathbf{e}}_i^1$ is that same edge shifted half a cell width in the x_1-direction. We do not need a notation for the edges $\mathbf{e}_i^2$ and $\mathbf{e}_i^3$ shifted in the x_1-direction, as these do not appear in the numerical schemes. The *bi-axially shifted grids* and their corresponding nodes, edges, faces and volumes are denoted by $\left(\widetilde{\widetilde{\mathcal{G}}}^j, \widetilde{\widetilde{\mathbf{n}}}_i^j, \widetilde{\widetilde{\mathbf{e}}}_i^j, \widetilde{\widetilde{\mathbf{f}}}_i^j, \widetilde{\widetilde{\mathbf{v}}}_i^j\right)$ where the upper index j indicates the direction along which the grid is not shifted (Fig. 3.1c). Note again the double meaning of the upper index. For example $\widetilde{\widetilde{\mathbf{f}}}_i^3$ is the face $\mathbf{f}_i^3$ shifted half a cell width along the x_1- and x_2-axes. The equally shifted versions of $\mathbf{f}_i^1$ and $\mathbf{f}_i^2$ do not appear in the numerical schemes. Hence no notation is introduced for them. The *tri-axially shifted grid*, better known as the dual grid, and its corresponding nodes, edges, faces and volumes are denoted by $\left(\widetilde{\widetilde{\widetilde{\mathcal{G}}}}, \widetilde{\widetilde{\widetilde{\mathbf{n}}}}_i, \widetilde{\widetilde{\widetilde{\mathbf{e}}}}_i^j, \widetilde{\widetilde{\widetilde{\mathbf{f}}}}_i^j, \widetilde{\widetilde{\widetilde{\mathbf{v}}}}_i\right)$ where the upper index j indicates the direction of the edges and faces (Fig. 3.1d).

Table 3.1 summarises the relations between the mesh elements that have just been introduced. The first row displays all nodes of the different staggered grids: the primary node $\mathbf{n}_{i_1,i_2,i_3}$, the three uni-axially shifted nodes, the three bi-axially shifted nodes and the tri-axially shifted node. Rows two, three and four hold the x_1-, x_2- and x_3-edges respectively. Rows four, five and six hold the x_3-, x_2- and x_1-faces. In the bottom row all volumes are listed. The organisation of the table entries is such that all elements of one column are co-located, i.e, the centre of all these elements coincide. The diagonal holds the mesh elements of the primary grid. The anti-diagonal holds all mesh elements of the dual grid.

The mesh elements that should be in the blank cells are not named, since these are not required for this discussion.

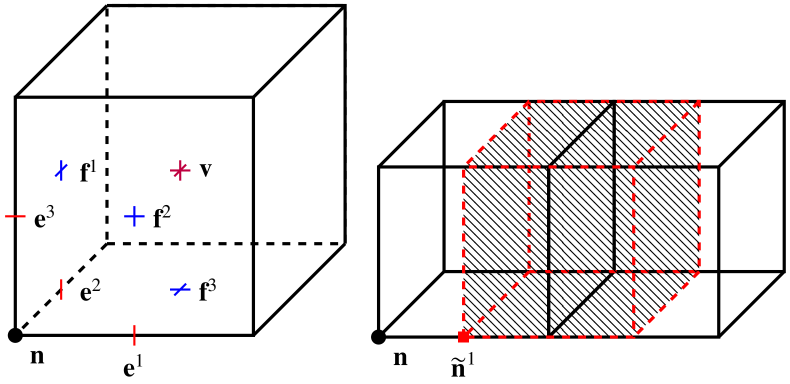
3.1.2 Application of the finite integration technique

The continuous problems are discretised in space by applying the integral formulations of the equations in systems (2.8), (2.10) and (2.12) on appropriate mesh elements, in particular facets and cells, of the computational domain. E.g., the second equation of (2.10) is typically integrated over primary facets, while the second equation of (2.12) is integrated over uni-axially shifted cells. The choice of which mesh elements are used for which fields, is guided by the requirement that the discrete fields should have the same continuity properties as the continuous fields. The subsequent expansion of the resulting integrals, leads to central difference stencils and hence a second order accurate spatial discretisation.

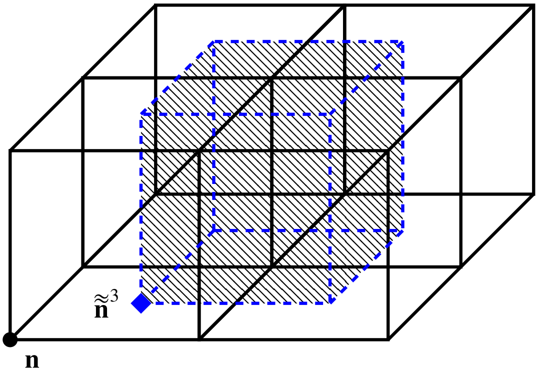
The procedure described in the preceding paragraph localises the components of the unknown fields at certain positions in a cell. The unique stencils for the different problems, as summarised by Marklein [70], are shown in Figure 3.2. Despite the uniqueness of these stencils, there is still a choice at which mesh elements the dofs should be allocated, as will be discussed in Section 3.2.2. The different components of the unknown fields are not co-located, i.e., they are represented at different mesh elements. As a consequence, material parameters need to be available at different mesh elements. This will result in non-co-location of material information, which will be discussed in Section 3.2.4. The scheme with non-co-located unknown field components guarantees appropriate interface conditions at material jumps by construction.

$\mathbf{n}_{i_1,i_2,i_3}$	$\widetilde{\mathbf{n}}^1_{i_1,i_2,i_3}$	$\widetilde{\mathbf{n}}^2_{i_1,i_2,i_3}$	$\widetilde{\mathbf{n}}^3_{i_1,i_2,i_3}$	$\widetilde{\widetilde{\mathbf{n}}}^3_{i_1,i_2,i_3}$	$\widetilde{\widetilde{\mathbf{n}}}^2_{i_1,i_2,i_3}$	$\widetilde{\widetilde{\mathbf{n}}}^1_{i_1,i_2,i_3}$	$\widetilde{\widetilde{\widetilde{\mathbf{n}}}}_{i_1,i_2,i_3}$
$\widetilde{\mathbf{e}}^1_{\underline{i_1},i_2,i_3}$	$\mathbf{e}^1_{i_1,i_2,i_3}$					$\widetilde{\widetilde{\mathbf{e}}}^1_{\underline{i_1},i_2,i_3}$	$\widetilde{\widetilde{\mathbf{e}}}^1_{i_1,i_2,i_3}$
$\widetilde{\mathbf{e}}^2_{i_1,\underline{i_2},i_3}$		$\mathbf{e}^2_{i_1,i_2,i_3}$			$\widetilde{\widetilde{\mathbf{e}}}^2_{i_1,\underline{i_2},i_3}$		$\widetilde{\widetilde{\mathbf{e}}}^2_{i_1,i_2,i_3}$
$\widetilde{\mathbf{e}}^3_{i_1,i_2,\underline{i_3}}$			$\mathbf{e}^3_{i_1,i_2,i_3}$	$\widetilde{\widetilde{\mathbf{e}}}^3_{i_1,i_2,\underline{i_3}}$			$\widetilde{\widetilde{\mathbf{e}}}^3_{i_1,i_2,i_3}$
$\widetilde{\widetilde{\mathbf{f}}}^3_{\underline{i_1},\underline{i_2},i_3}$			$\widetilde{\widetilde{\mathbf{f}}}^3_{\underline{i_1},\underline{i_2},i_3}$	$\mathbf{f}^3_{i_1,i_2,i_3}$			$\widetilde{\mathbf{f}}^3_{i_1,i_2,i_3}$
$\widetilde{\widetilde{\mathbf{f}}}^2_{\underline{i_1},i_2,\underline{i_3}}$		$\widetilde{\widetilde{\mathbf{f}}}^2_{\underline{i_1},i_2,\underline{i_3}}$			$\mathbf{f}^2_{i_1,i_2,i_3}$		$\widetilde{\mathbf{f}}^2_{i_1,i_2,i_3}$
$\widetilde{\widetilde{\mathbf{f}}}^1_{i_1,\underline{i_2},\underline{i_3}}$	$\widetilde{\widetilde{\mathbf{f}}}^1_{i_1,\underline{i_2},\underline{i_3}}$					$\mathbf{f}^1_{i_1,i_2,i_3}$	$\widetilde{\mathbf{f}}^1_{i_1,i_2,i_3}$
$\widetilde{\widetilde{\widetilde{\mathbf{v}}}}_{\underline{i_1},\underline{i_2},\underline{i_3}}$	$\widetilde{\widetilde{\mathbf{v}}}^1_{i_1,\underline{i_2},\underline{i_3}}$	$\widetilde{\widetilde{\mathbf{v}}}^2_{\underline{i_1},i_2,\underline{i_3}}$	$\widetilde{\widetilde{\mathbf{v}}}^3_{\underline{i_1},\underline{i_2},i_3}$	$\widetilde{\mathbf{v}}^3_{i_1,i_2,\underline{i_3}}$	$\widetilde{\mathbf{v}}^2_{i_1,\underline{i_2},i_3}$	$\widetilde{\mathbf{v}}^1_{\underline{i_1},i_2,i_3}$	$\mathbf{v}_{i_1,i_2,i_3}$

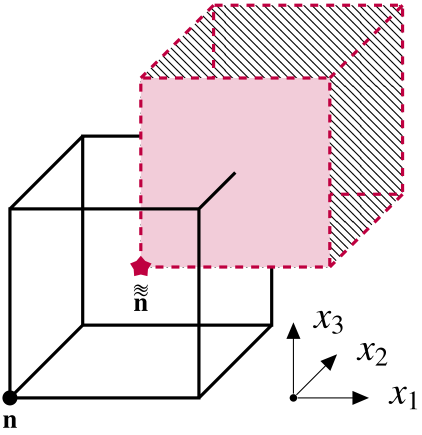
Table 3.1: Overview of the relation between the nodes, edges, faces and volumes of the primary, uni-, bi- and tri-axially shifted grids, as introduced in Section 3.1. We use the underlined indices $\underline{i_j} = i_j - 1$ for a more compact notation.



(a) Primary cell. (b) Uni-axially x_1-shifted cell.

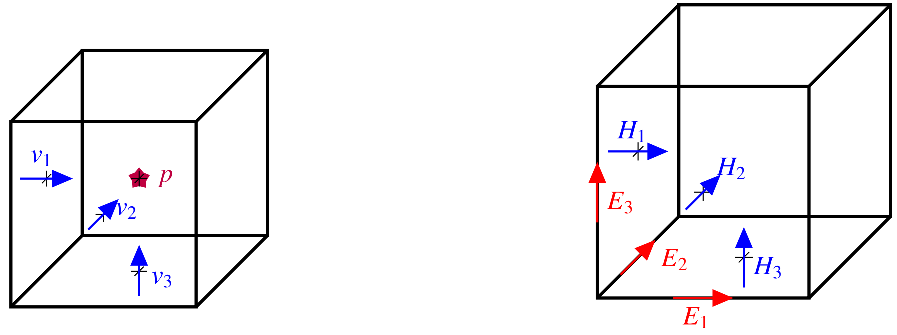


(c) Bi-axially x_1, x_2-shifted cell.



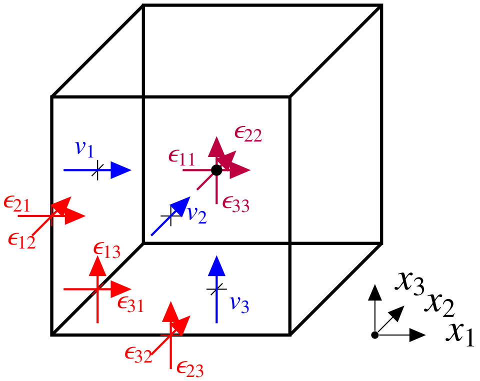
(d) Tri-axially shifted cell, also known as the dual cell.

Figure 3.1: Ensemble of shifted grids.



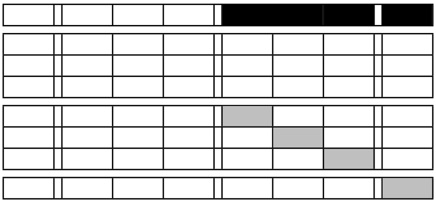
(a) Acoustic stencil, where p is the pressure and v_i are the components of the particle velocity as defined in (2.8).

(b) Electromagnetic stencil, where E_i and H_i are the components of the electric and magnetic field strength respectively as defined in (2.10).

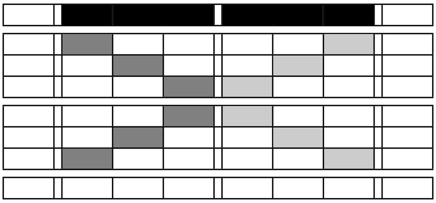


(c) Elastodynamic stencil, where v_i are the components of the particle velocity as defined in (2.12) and ϵ_{ij} are the components of the strain as defined in (2.15).

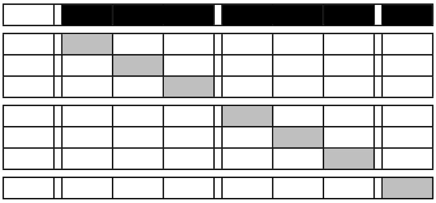
Figure 3.2: Unique stencils indicating the component locations of the unknown fields for the different problems.



(a) Acoustic stencil.



(b) Electromagnetic stencil.



(c) Elastodynamic stencil.

Figure 3.3: Tables (a), (b) and (c) act as masks for Table 3.1 and indicate the nodal allocations (black) and mimetic allocations (grey) (see Section 3.2.2) for AFIT, EMFIT and EFIT respectively. For EMFIT (b), two mimetic allocations are indicated, depending on the chosen tensor fields to describe the problem: the electric field strength $\boldsymbol{E}$ and the magnetic field strength $\boldsymbol{H}$ (dark grey) or the electric displacement $\boldsymbol{D}$ and the magnetic induction $\boldsymbol{B}$ (light grey).

3.2 Semi-discrete problem

The use of FIT to derive the semi-discrete formulation of the acoustic, electromagnetic and elastic wave problem, leads to a matrix equation of the form

$$\begin{pmatrix} \mathbf{M}_u & 0 \\ 0 & \mathbf{M}_v \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{v}} \end{pmatrix} = \begin{pmatrix} 0 & -\mathbf{K}^T \\ \mathbf{K} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} + \begin{pmatrix} 0 \\ \boldsymbol{j} \end{pmatrix}, \tag{3.1}$$

with $\boldsymbol{u}$ and $\boldsymbol{v}$ vectors containing the dofs. The matrices $\mathbf{M}_u$ and $\mathbf{M}_v$ are discrete Hodge operators, i.e., the discrete counterparts of the material relations (see Section 3.2.3). The matrices $\mathbf{K}$ and $\mathbf{K}^T$ are discrete topological operators (gradient, divergence, vector gradient, vector divergence, curl), which will be discussed in Section 3.2.1. Vector $\boldsymbol{j}$ contains the discrete source field. The discretisation by FIT for all three problems has been described in detail in [16, 70, 98].

In the following subsections, we introduce a general description of how the dofs in System (3.1) are defined and how the matrices are constructed. We provide and substantiate several properties that justify the name *mimetic allocation*. First, the discrete topological operators preserve the properties of their continuous counterparts exactly. Secondly, every continuous integral relation has an exact discrete counterpart. Finally, the existence of a discrete conservation law implies stability of the discretisation.

The generic description that we introduce here, shows that the difference between the considered allocations, and hence between FIT and FDM, is only a matter of where the metric information of the problem, i.e., edge lengths, face surfaces and cell volumes, enters System (3.1). Hence both methods can be called mimetic discretisation methods when applied to the considered problem. This shared classification allows a unified numerical analysis.

3.2.1 Topological operators

The matrices $\mathbf{K}$ and $\mathbf{K}^T$ in (3.1) are discrete counterparts of topological operators. Every row of $\mathbf{K}$ and $\mathbf{K}^T$ contains a few 1 and -1 elements as a result of difference stencils. Details about the construction of these matrices are given in [87]. In total, there are ten topological matrices, since all five operators have both a primary and a dual discrete counterpart. As shown in [18, 56, 55, 70], these discrete operators have the same properties as their continuous equivalents, which is a first requirement for mimetic discretisations. Depending on the allocation of the dofs, the pure topological matrices may have to be multiplied by a diagonal matrix containing sizes of certain mesh elements, as is done in [16, 70].

3.2.2 Degrees of freedom

Degrees of freedom can be associated with the nodes of the set of staggered grids (*nodal allocation*) or with other mesh elements of the set of staggered grids (*mimetic allocation*). Nodal allocation is common for acoustic and elastic wave propagation whereas mimetic allocation is typical for electromagnetic problems. As a key point in this chapter, we show in Figure 3.3 that both allocations are applicable to all three problems in a unified way and we indicate the equivalence of both allocations.

The mimetic allocation defines the dofs on mesh elements according to the theory of differential forms [90, 93]. This choice results in a discrete equivalent of the integral theorems of vector calculus (Green, Gauss, Stokes), which is a second desired property of mimetic discretisations [18, 56]. The use of non-nodal dofs results in *pure* topological operators, i.e., sparse matrices with only elements from the set $\{-1, 0, 1\}$ in a structured way, which leads to an efficient implementation. The nodal allocation approach, most frequently used in acoustics and elastodynamics [16, 70], complies with the traditional FDM discretisations, but results in more complex topological operators.

Mimetic dofs allocated at mesh elements, are transferred into nodal dofs by division by the size of the corresponding mesh element. As a consequence, the systems resulting from nodal discretisations as, e.g., FDM, only differ from those related to the corresponding mimetic discretisation by a shift of metric information from the Hodge operators $\mathbf{M}_u$ and $\mathbf{M}_v$ and the topological operators $\mathbf{K}$ and $\mathbf{K}^T$ to the dofs $\boldsymbol{u}$ and $\boldsymbol{v}$.

3.2.3 Hodge operators

The discrete Hodge operators $\mathbf{M}_u$ and $\mathbf{M}_v$ have been studied thoroughly in the electromagnetic case [19, 52]. Rather than describing these matrices for the three problems and for both allocations one by one, we take a more unified approach. Consider a material equation $\boldsymbol{f}(\boldsymbol{x}) = \alpha \boldsymbol{g}(\boldsymbol{x})$ with nth order tensor fields $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$ and a tensorial material parameter α of appropriate order. We denote the dofs of $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ by $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively. They are allocated on mesh elements with the same centre node. The discretisation of the material equation at a centre node i is then of the form

$$\frac{\boldsymbol{a}_i}{\mathcal{E}\left(\boldsymbol{a}_i\right)} = \alpha_i \frac{\boldsymbol{b}_i}{\mathcal{E}(\boldsymbol{b}_i)}, \tag{3.2}$$

The operator $\mathcal{E}\left(\boldsymbol{a}\right)$ indicates *the size of the mesh element at which dof* $\boldsymbol{a}$ *is allocated.* The definition and allocation of the material parameters α_i will be discussed in Section 3.2.4. As an example we consider the electromagnetic material relation

$$\boldsymbol{B} = \mu \boldsymbol{H}, \tag{3.3}$$

where $\boldsymbol{B}$ is the magnetic induction, μ is the permeability and $\boldsymbol{H}$ is the magnetic field strength. The dofs $\boldsymbol{b}$ of $\boldsymbol{B}$ are allocated on primary facets and the dofs $\boldsymbol{h}$ of $\boldsymbol{H}$ on dual edges (see Figure 3.2b). These allocations are indicated in columns five, six and seven of Figure 3.3b. For this case, relation (3.2) becomes

$$\frac{\boldsymbol{b}_i}{A_i} = \mu \frac{\boldsymbol{h}_i}{l_i}, \tag{3.4}$$

where A_i is the surface of the primary facet at which $\boldsymbol{b}_i$ is allocated and l_i is the length of the corresponding dual edge at which $\boldsymbol{h}_i$ is allocated. This material relation is expressed at bi-axially shifted nodes because those coincide with the component locations of both $\boldsymbol{B}$ and $\boldsymbol{H}$. For models with isotropic material distributions, the inverse of these operators can also be obtained in a similar way. This will be useful, for example, in combination with explicit time integration schemes.

3.2.4 Allocation of the material parameters

There are several possibilities for the allocation of the material parameters. We use a conformal material grid as primary grid, i.e., every primary cell is homogeneously filled with material and material jumps are located at primary facets. The material information of each cell is represented by one tensorial material parameter, which is allocated at the centre of the primary cell, i.e., the dual nodes. When the dofs are allocated at shifted nodes, the needed material parameters have to be averaged out to the shifted nodes at which the dofs are allocated. We will use the shifted cells for the averaging process. It is however possible to choose other grids, e.g., rotated grids [84].

The material parameter $\alpha_{\mathcal{N}}$ allocated at a shifted node $\mathcal{N}$ is obtained from the primary mesh parameters by volumetric averaging using shifted cells as follows

$$\alpha_{\mathcal{N}} = \frac{\displaystyle\sum_{i=1}^{N} \alpha_{\tilde{\tilde{\tilde{\mathbf{n}}}}_i} \left|\mathcal{V} \cap \mathbf{v}_i\right|}{\left|\mathcal{V}\right|},$$

where $\mathcal{V}$ is the dual cell of node $\mathcal{N}$, i.e., the cell that is mentioned in the same column as $\mathcal{N}$ in Table 3.1, $\mathbf{v}_i$ is the cell of which $\overset{\approx}{\tilde{\mathbf{n}}}_i$ is the primary centre and $|\nu|$ is the volume of the domain ν.

3.2.5 Discrete energy

Schuhmann and Weiland [87] showed that the conservation of the discrete energy is a necessary and sufficient condition for stability of the spatial discretisation scheme. This is a third desired property for a discretisation scheme to be considered a mimetic method.

The discrete energy E in the system can be computed as

$$E = \frac{1}{2}\left(\boldsymbol{u}^T \mathbf{M}_u \boldsymbol{u} + \boldsymbol{v}^T \mathbf{M}_v \boldsymbol{v}\right). \tag{3.5}$$

3.3 Time integration

Apart from the spatial discretisation of the continuous problem, the semi-discrete problem has to be discretised in time, and integrated over time. This is realised by applying time stepping schemes. As we proposed a spatial discretisation method that can provide inverse Hodge operators, it makes sense to use an explicit time stepping scheme. It is of course possible to use implicit methods, which can be advantageous in certain specific problems, e.g., electromagnetic problems with damping as shown by Verwer and Botchev [97]. In the following subsections, we first recall the classical second order leapfrog method, followed by the introduction of a second order composition method. Next, the composition method is generalised to higher order variants [50, pp. 152-158].

3.3.1 Classical leapfrog method

The leapfrog method is by far the most commonly used method to solve wave propagation problems [89]. This method is also known under different names, such as the Yee-scheme [101] or the Störmer-Verlet-scheme [49]. The scheme computes the dofs of the various tensor fields at different time instants; it is conditionally stable and has second order accuracy:

$$\mathbf{M}_u \frac{\boldsymbol{u}^{n+1/2} - \boldsymbol{u}^{n-1/2}}{\Delta t} = -\mathbf{K}^T \boldsymbol{v}^n, \tag{3.6}$$

$$\mathbf{M}_v \frac{\boldsymbol{v}^{n+1} - \boldsymbol{v}^n}{\Delta t} = \mathbf{K}\boldsymbol{u}^{n+1/2} + \boldsymbol{j}^{n+1/2}, \tag{3.7}$$

where Δt is the time step size.

3.3.2 Second order composition method

The second order composition method or CO2 method, see Hairer et al. [50, Section III.5.4], was first used as a time integration method for solving the Maxwell wave equation by Botchev and Verwer in [21, 97]. The main difference with the leapfrog method is that the dofs of both unknown fields are available at full time steps, which is advantageous for calculating the discrete energies of the problem. The calculation rule is

$$\mathbf{M}_u \frac{\boldsymbol{u}^{n+1/2} - \boldsymbol{u}^{n}}{\Delta t/2} = -\mathbf{K}^T \boldsymbol{v}^n, \tag{3.8}$$

$$\mathbf{M}_v \frac{\boldsymbol{v}^{n+1} - \boldsymbol{v}^n}{\Delta t} = \mathbf{K}\boldsymbol{u}^{n+1/2} + \frac{1}{2}\left(\boldsymbol{j}^{n} + \boldsymbol{j}^{n+1}\right), \tag{3.9}$$

$$\mathbf{M}_u \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n+1/2}}{\Delta t/2} = -\mathbf{K}^T \boldsymbol{v}^{n+1}, \tag{3.10}$$

for which one additional equation has to be solved compared to the leapfrog scheme. However, Equation (3.8) is nearly identical to Equation (3.10) of the previous time step. By storing the vector $-\frac{\Delta t}{2}{\mathbf{M}_u}^{-1}\mathbf{K}^T\boldsymbol{v}^{n+1}$ in a temporary variable in the last stage, and reusing this in the first stage of the next step, the additional computational cost is limited to one vector addition.

As noted in Botchev and Verwer [21, Remark 3.1], by substituting (3.10) into (3.8) the classical leapfrog scheme is retrieved. The benefit of computing this time step by solving two halve time steps is that both fields are available at full time steps, allowing a more precise computation of the discrete energy in the system. We will regard both methods as equivalent and use CO2 as the reference method.

3.3.3 Higher order composition methods

The CO2 method can be generalised to higher order schemes. In general, the time stepping can be described by s stages as follows.

Set $U_0 = \boldsymbol{u}^n$ en $V_0 = \boldsymbol{v}^n$, and solve

$$\mathbf{M}_u \frac{U_k - U_{k-1}}{\Delta t} = -\left(\beta_k + \alpha_{k-1}\right) \mathbf{K} V_{k-1}, \tag{3.11}$$

$$\mathbf{M}_v \frac{V_k - V_{k-1}}{\Delta t} = \left(\beta_k + \alpha_k\right) \mathbf{K}^T U_k - \boldsymbol{j}(t_k^u), \tag{3.12}$$

for k from 1 to s, followed by a final step

$$\mathbf{M}_u \frac{\boldsymbol{u}^{n+1} - U_s}{\Delta t} = -\alpha_s \mathbf{K} \boldsymbol{v}^{n+1}; \qquad \boldsymbol{v}^{n+1} = V_s, \tag{3.13}$$

where the intermediate time instants t_k^u are defined as $t_k^u = t^n + \Delta t \left(\tilde{\alpha}_{k-1} + \tilde{\beta}_k\right)$, and the auxiliary variables are $\tilde{\alpha}_k = \sum_{i=0}^{k} \alpha_i$ and $\tilde{\beta}_k = \sum_{i=0}^{k} \beta_i$. The parameters α_i and β_i to construct 4th, 6th, 8th and 10th order methods are listed in Hairer et al. [50, pp. 152-158]. In this chapter we will consider the 4th (CO4) and 6th (CO6) order composition rules.

3.4 Efficiency study for time stepping methods applied in combination with FIT

In this section we present simulation results for wave problems in acoustics, electromagnetics and elastodynamics using FIT. We do this by considering three academic models, for which an analytical solution is available, and two technically relevant problems. All academic models use an equidistant mesh on a unit cube ($[0,1]^3$ [m^3]). First we use the models to verify the accuracy order of the presented time integration methods by solving the problem on a certain time interval $[0,T]$ and refining the time step size, while keeping the mesh fixed. Secondly, we compare the efficiency of the different methods for every model. For the academic models, the analytical solution is used as the reference solution to compute the error. For the technically relevant problems, no analytical solution is known. For the first application, the results of commercial software CST [32] is used as reference. For the second application, we verify the results of a certain time integrator by using the results of this method computed with a time step size which is smaller than the finest displayed time step, as a reference solution.

The discussed time integration methods will be compared to the classical four stage fourth order Runge-Kutta method (RK4).

In the following subsection, we introduce the three academic models. In Sections 3.4.4 and 3.4.5, an electromagnetic and elastodynamic application are introduced. An accuracy and efficiency plot will be presented for all introduced problems. Afterwards, all results will be discussed in Section 3.5. The purpose of the efficiency plots that will be presented is to visualise which method is faster to obtain a certain desired accuracy. As we are investigating the accuracy and efficiency of the time integrators, we choose a large number of cells per wavelength to eliminate side-effects caused by a too large spatial discretisation error.

3.4.1 Acoustic model

For the acoustic model we consider a unit cube of air ($K = 101000$ Pa, $\rho = 1.269$ kg/m^3) and a volume force density $\boldsymbol{f} = (f_1, f_2, f_3)$, where the components are

$$f_1 = \left(K\frac{k_1^2}{\omega_1} - \rho\omega_1\right)\cos\left(k_1 x - \omega_1 t\right), \tag{3.14}$$

$$f_2 = \left(K\frac{k_2^2}{\omega_2} - \rho\omega_2\right)\cos\left(k_2 y - \omega_2 t\right), \tag{3.15}$$

$$f_3 = \left(K\frac{k_3^2}{\omega_3} - \rho\omega_3\right)\cos\left(k_3 z - \omega_3 t\right), \tag{3.16}$$

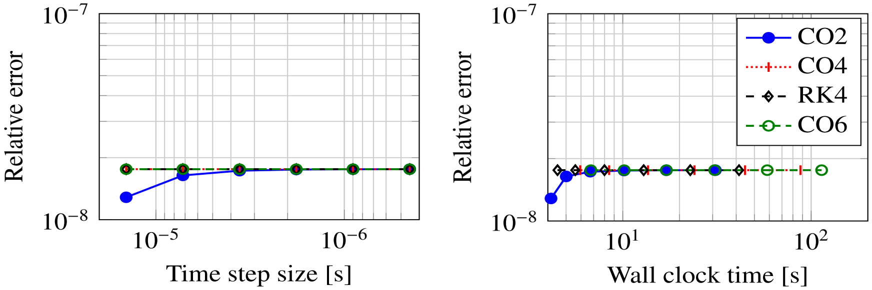
and $\boldsymbol{k} = (k_1, k_2, k_3)$ is the wave number and ω_1, ω_2, ω_3 are the angular wave frequencies. The analytical solution for the particle velocity $\boldsymbol{v} = (v_1, v_2, v_3)$ is

$$v_1 = \sin\left(k_1 x - \omega_1 t\right), \quad v_2 = \sin\left(k_2 y - \omega_2 t\right), \quad v_3 = \sin\left(k_3 z - \omega_3 t\right),$$

and the analytical solution for the pressure is

$$p(x) = K\frac{k_1}{\omega_1}\sin\left(k_1 x - \omega_1 t\right) + K\frac{k_2}{\omega_2}\sin\left(k_2 y - \omega_2 t\right) + K\frac{k_3}{\omega_3}\sin\left(k_3 z - \omega_3 t\right).$$

The angular velocities are chosen to be $\omega_1 = 1$, $\omega_2 = 10$, $\omega_3 = 100$ rad/s. The used equidistant mesh with cell width $\Delta x = \Delta y = \Delta z = 2^{-6}$ m assures at least 1100 cells per wavelength. The normal components of the velocity at the boundary of the computational domain are set equal to the normal components of the analytical solution for the particle velocity. Figures 3.4a and 3.4b show the accuracy and efficiency plots for this model.



(a) AFIT convergence.

(b) AFIT timings.

Figure 3.4: Acoustic model (AFIT). Relative error of the time discretisation compared to the analytic solution as a function of (a) the time step size and (b) the wall clock time.

3.4.2 Electromagnetic model

The electromagnetic model is borrowed from Botchev and Verwer [21] and concerns a forced vibration in a unit cube of vacuum ($\varepsilon = 8.85\,10^{-12}$ F/m, $\mu = 1.26\,10^{-6}$ H/m). The source field is given by:

$$\boldsymbol{J}_s(x,y,z,t) = -\varepsilon\dot{\alpha}(t)\boldsymbol{E}_{\text{stat}}(x,y,z) + \beta(t)\nabla\times\boldsymbol{H}_{\text{stat}}(x,y,z),$$

where the used scalar time-dependent functions are defined as

$$\alpha(t) = \sum_{k=1}^{n}\cos(\omega_k t), \qquad \beta(t) = -\frac{1}{\mu}\sum_{k=1}^{n}\frac{\sin(\omega_k t)}{\omega_k}. \tag{3.17}$$

The analytical solutions are

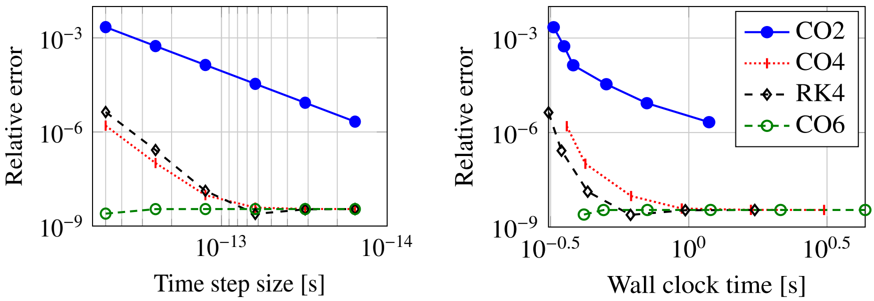
$$\boldsymbol{E}(x,y,z,t) = \alpha(t)\boldsymbol{E}_{\text{stat}}(x,y,z), \tag{3.18}$$

$$\boldsymbol{H}(x,y,z,t) = \beta(t)\boldsymbol{H}_{\text{stat}}(x,y,z), \tag{3.19}$$

$$\boldsymbol{E}_{\text{stat}}(x,y,z) = \big(\sin(\pi y)\sin(\pi z),\,\sin(\pi x)\sin(\pi z),\,\sin(\pi x)\sin(\pi y)\big)\,, \tag{3.20}$$

$$\boldsymbol{H}_{\text{stat}}(x,y,z) = \nabla\times\boldsymbol{E}_{\text{stat}}(x,y,z). \tag{3.21}$$

For this model, we choose $n = 3$ and the angular wave frequencies $\omega_k = \frac{10^{12}}{k}$ rad/s. An equidistant mesh with cell width $\Delta x = \Delta y = \Delta z = 2^{-4}$ is used. As boundary conditions, the tangential components of the electric field are set



(a) EMFIT convergence.

(b) EMFIT timings.

Figure 3.5: Electromagnetic model (EMFIT). Relative error of the time discretisation compared to the analytic solution as a function of (a) the time step size and (b) the wall clock time.

equal to the tangential components of the analytical solution for the electric field. Figures 3.5a and 3.5b show the accuracy and efficiency plots for this model.

3.4.3 Elastodynamic model

The elastodynamic model concerns a unit cube of aluminium ($\rho = 2700$ kg/m^3, Poisson's ratio $\nu = 0.33$, shear modulus $G = 26.32$ [GPa]) with an applied volume force density field $\boldsymbol{f} = (f_1, f_2, f_3)$, where

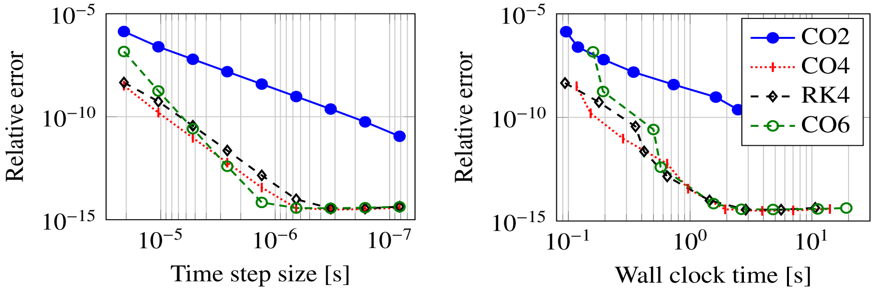
$$f_1 = -\frac{\rho\omega_x^2 - Gk_1^2}{\omega_1}\cos\left(k_1 z - \omega_1 t\right), \tag{3.22}$$

$$f_2 = -\frac{\rho\omega_y^2 - Gk_2^2}{\omega_2}\cos\left(k_2 x - \omega_2 t\right), \tag{3.23}$$

$$f_3 = -\frac{\rho\omega_z^2 - Gk_3^2}{\omega_3}\cos\left(k_3 y - \omega_3 t\right). \tag{3.24}$$

The analytical solution for the particle velocity $\boldsymbol{v} = (v_1, v_2, v_3)$ is

$$v_1 = \sin\left(k_1 z - \omega_1 t\right), \quad v_2 = \sin\left(k_2 x - \omega_2 t\right), \quad v_3 = \sin\left(k_3 y - \omega_3 t\right), \tag{3.25}$$



(a) EFIT convergence.

(b) EFIT timings.

Figure 3.6: Elastodynamic model (EFIT). Relative error of the time discretisation compared to the analytic solution as a function of (a) the time step size and (b) the wall clock time.

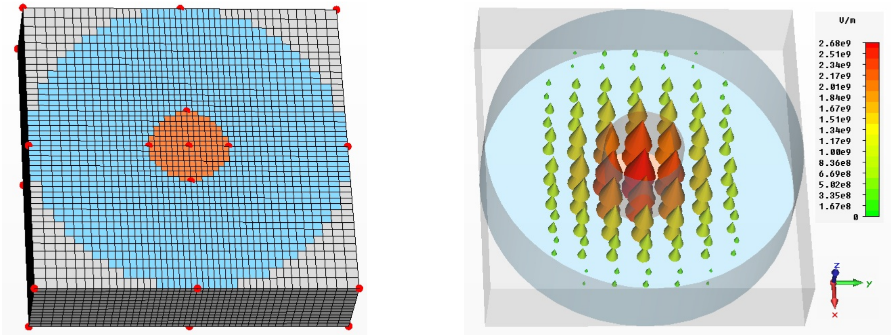
and the analytical solution for the stress $\boldsymbol{\sigma}$ is

$$\begin{aligned}
\sigma_{11} &= \sigma_{22} = \sigma_{33} = 0, \\
\sigma_{23} &= \sigma_{32} = -\frac{Gk_3}{\omega_3}v_3, \\
\sigma_{13} &= \sigma_{31} = -\frac{Gk_1}{\omega_1}v_1, \\
\sigma_{12} &= \sigma_{21} = -\frac{Gk_2}{\omega_2}v_2.
\end{aligned} \tag{3.26}$$

The wave numbers are chosen to be $(k_1, k_2, k_3) = (\pi, \pi, \pi)10^{-2}$ m^{-1}. The used equidistant mesh with cell width $\Delta x = \Delta y = \Delta z = 2^{-3}$ m assures at least 800 cells per wavelength. As boundary conditions, the shear stresses at the boundaries are set to the values of the analytical solution for the shear stresses. Figures 3.6a and 3.6b show the accuracy and efficiency plots for this model.

3.4.4 Electromagnetic application

In this section, the first eigenfrequency of a resonating cavity is calculated. We consider a cylindrical cavity with radius 4 mm and height 2.5 mm. In this vacuum cylinder ($\varepsilon = \varepsilon_0$, $\mu = \mu_0$), a stem is placed with radius 1 mm, permittivity $\varepsilon = 4\varepsilon_0$ and permeability $\mu = \mu_0$. A perfect electric boundary condition (PEC) is set on the full surface of the cavity. The reference solution of 22 GHz is obtained with the eigenmode solver of the commercial software *CST MICROWAVE STUDIO* [32]. The used mesh and first eigenmode are shown in Figure 3.7. We search for eigenfrequencies in the range from 1 GHz to 100 GHz, since the maximal cell width in the used mesh is 0.148 mm, there are at least 20 cells per wavelength. For the expected eigenfrequency of 22 GHz there are at least 80 cells per wavelength, which is more than sufficient to suppress numerical dispersion.



(a) Mesh of the resonator.

(b) First eigenmode (at 22 GHz) of the resonator.

Figure 3.7: Mesh (a) and first eigenmode (b) of a resonating cylindrical cavity with stem.

A modulated Gaussian pulse in space and time is applied to the centre of the mesh, namely $Js_1 = Js_2 = Js_3 =$

$$\sin(2\pi f_{\text{peak}} t) \exp\left(-(t\pi f_{\text{peak}})^2\right) \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}{\sigma}\right), \tag{3.27}$$

where $f_{\text{peak}} = 50$ GHz is the peak frequency, $(x_0, y_0, z_0) = (0, 0, 1.25)$ mm is the centre of the pulse and $\sigma = 1$ mm determines the width of the pulse. After using CO2 for 50000 time steps with time step size $\Delta t = \frac{10^{-11}}{512}$ s, the Fourier transform of the electric field $\boldsymbol{E}$ at (x_0, y_0, z_0) is taken. The plot of this result is shown in Figure 3.8. The peaks in the spectrum indicate the resonating frequencies.

The smallest eigenvalue is the most important one and is found at 21.5 GHz. The corresponding eigenmode consists of an axial electric field and an azimuthal magnetic field. This matches the results obtained from *CST MICROWAVE STUDIO* as shown in Figure 3.7b.

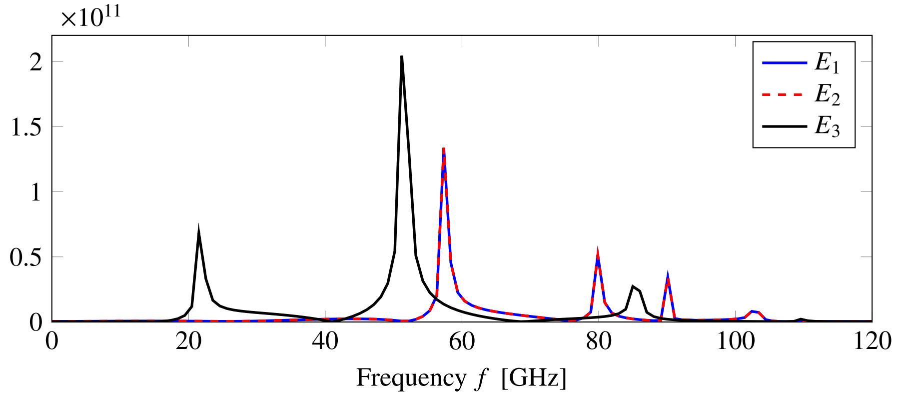


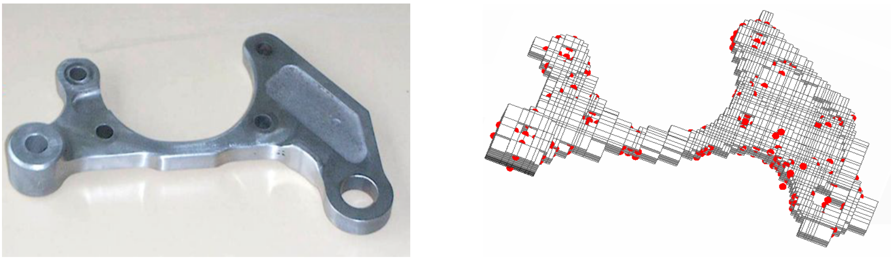
Figure 3.8: Spectrum of all components of the electric field $\boldsymbol{E} = (E_1, E_2, E_3)$ at $(x_0, y_0, z_0) = (0, 0, 1.25)$ mm in the cylindrical resonator shown in Figure 3.7. The first peak is located at $f = 21.5$ GHz, which is up to 2 digits accurate. Also the other peaks match eigenfrequencies found with *CST MICROWAVE STUDIO* with the same accuracy. Note that the spectra of E_1 and E_2 coincide due to the symmetry of the geometry. The height of the peaks is irrelevant in this procedure.

3.4.5 Elastodynamic application

The final example concerns a steel bracket (mass density $\rho = 7850$ kg/m^3, Poisson's ratio $\nu = 0.29$, shear modulus $G = 77.52$ [GPa]) that is part of a landing gear of an aircraft (Fig. 3.9a). This device has been the subject of non-destructive testing experiments to detect defects by Van Damme et al. [95]. For this model, we use EFIT on a non-equidistant Cartesian tensor-product grid. The boundaries are modelled to be stress free. The source field is again a modulated Gaussian pulse as introduced in equation (3.27) with $f_{\text{peak}} = 500$ Hz applied at an arbitrary position in the bracket. All initial conditions are set to zero. The used mesh provides at least 100 cells per wavelength for $f_{\max} = 1$ kHz. Figures 3.10a and 3.10b show the accuracy and efficiency plots for this model.

3.5 Discussion

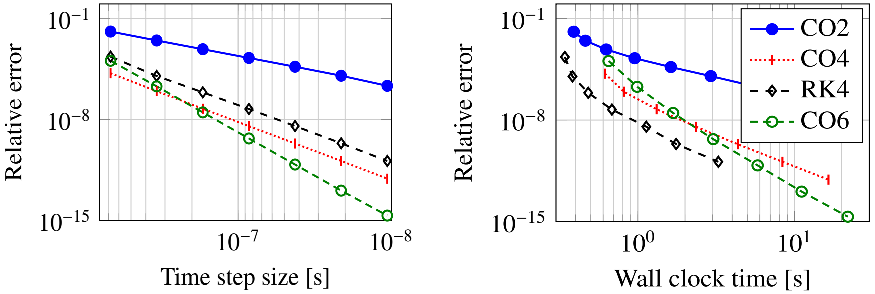
The coarsest time step size used in all simulations is the largest possible that fulfils the CFL condition [31]. In the AFIT convergence plot (Fig. 3.4a), we see that all time integration methods reach the spatial discretisation error even with the coarsest time step size. This suggests that there is no benefit from using



(a) Bracket.

(b) Non-equidistant Cartesian tensor-product grid of the bracket.

Figure 3.9: Bracket (a) and the non-equidistant Cartesian tensor-product grid that was used for the simulations (b).



(a) Bracket convergence.

(b) Bracket timings.

Figure 3.10: Elastodynamic application (bracket). Relative error of the time discretisation, compared with a solution obtained by using a time step size which is half the size of the smallest displayed time step size, as a function of (a) the time step size and (b) the wall clock time.

higher order methods. In this case it might be advantageous to use implicit time integration methods in order to be able to use larger time step sizes. For the other two cases, we observe that the coarsest time step size does not necessarily lead to the most accurate solution when the second order method is used. Hence, the use of higher order methods, e.g., RK4, can be advantageous when coarse time steps are considered. We note that in some cases a larger time step size results in a more accurate solution as e.g., CO2 in Figure 3.4a. This behaviour is not surprising due to the resemblance with the leapfrog and Störmer methods, which can suffer from small instabilities for small step sizes as argued by Hairer et al. [49, III.10].

The convergence plot for the elastodynamic application (Fig. 3.10a) shows that the theoretical accuracy order of all methods is achieved. For the academic models, we see that the error stagnates before double precision is reached. This is due to the fact that these error curves include both the error of the spatial discretisation and the error of the time integration. In the case of the bracket model, the spatial discretisation error is not incorporated, as the reference solution is computed with the same spatial discretisation.

Looking at the efficiency plots Figures (3.4-3.6 and 3.10)(b), we can conclude that in general the CO2 method is slower than the higher order methods to reach highly accurate results. Hence, it can be advantageous to use higher order methods when simulating wave equation problems, even when only a low order spatial discretisation is used. From these results we learn that RK4 is at least as efficient as the fourth order composition method CO4. It is also not always better to use the highest possible order, since lower order methods may have a better error constant, as can be observed for the academic elastodynamic model in Figure 3.6. When results with a low accuracy are sufficient, the CO2 is equally fast as RK4.

The two presented applications suffer from the use of staircase meshes. In this chapter no effort has been made to overcome this problem, however many specific solutions can be fitted in the presented unified framework. Alternatively, unstructured meshes can be employed, which is the topic of the following chapter.

3.6 Conclusions

In this chapter, a general framework of eight staggered grids was presented. It was shown that the spatial discretisation of the acoustic, electromagnetic and elastic wave propagation all fit into this framework. Using this framework, we have established the relation between mimetic and nodal discretisations of these problems. Consequently, results from mimetic discretisation theory also hold for the classical approaches. This framework allows a unified application of the finite integration technique and the finite difference method to wave problems. In addition, it was shown that for these problems, if high accuracy is desired, it is time-efficient to use higher order time integration methods. If a fairly low accuracy is sufficient, the classical leapfrog method is as fast as higher order methods.

Chapter 4

Solving forward wave propagation problems using the finite element method

The finite element method (FEM) is a spatial discretisation method that can be used for unstructured grids. We consider both continuous and discontinuous finite elements [8]. FEM with continuous elements favors implicit time stepping due to mesh-wise solves, while the Discontinuous Galerkin FEM (DG-FEM) allows cell-wise solves and is more often used in combination with an explicit time integrator. For the remainder of this work, we use the acronym FEM exclusively when using the finite element method with continuous elements and DG-FEM when using the finite element method with discontinuous elements.

We use the FEniCS/DOLFIN software [6, 67, 68] for the implementation of finite element (FE) problems, which allows writing high level FEM code by means of the unified form language (UFL, [5]). We include example FEniCS code in this chapter as it gives insight into FEM and the used implementation.

In this chapter, we first derive an FE formulation for the one dimensional scalar wave equation (2.4) - (2.5) to demonstrate the use of FEM for wave problems. Next, DG-FEM is used for the three dimensional generic wave equation (2.7). In the subsequent sections FEM is applied to the acoustic, electromagnetic and elastic wave problems.

After introducing all spatial discretisations and time integrators, we discuss how to choose the right combination of methods for NDT&E purposes.

The FE formulation will be given for a computational domain Ω that is partitioned by N_k cells K_i so that $\Omega \approx \Omega_h = \cup_{1=1}^{N_k} K_i$.

4.1 Finite element formulation of the abstract one dimensional scalar wave equation using continuous elements

To obtain an FE formulation the continuous wave equation is first transformed into a weak continuous variational formulation. Next, a weak discrete variational formulation is derived. Finally, the weak discrete variational formulation is transformed to a semi-discrete linear system similar to the linear system (3.1) obtained with the finite integration technique in Chapter 3. For this last step, we rely on the FEniCS/DOLFIN software that will interpret the UFL formulation of the problem.

Continuous variational formulation

The variational form of the one dimensional scalar wave equation is obtained by multiplying the continuous equations (2.4) and (2.5) with a range of test functions p and q from the test spaces $\hat{V}$ and $\hat{U}$ resp. and integrating over the computational domain Ω:

$$\begin{aligned} \int_\Omega \tfrac{\partial u}{\partial t} \cdot p \,\mathrm{d}x &= \int_\Omega a\tfrac{\partial v}{\partial x} \cdot p \,\mathrm{d}x \qquad \forall p \in \hat{V}, \\ \int_\Omega \tfrac{\partial v}{\partial t} \cdot q \,\mathrm{d}x &= \int_\Omega b\tfrac{\partial u}{\partial x} \cdot q \,\mathrm{d}x + \int_\Omega f \cdot q \,\mathrm{d}x \qquad \forall q \in \hat{U}. \end{aligned} \tag{4.1}$$

Assuming the boundary conditions are given for u, we choose to integrate the second equation by parts, which leads to the weak variational form

$$\begin{aligned} \int_\Omega \tfrac{\partial u}{\partial t} \cdot p \,\mathrm{d}x &= \int_\Omega a\tfrac{\partial v}{\partial x} \cdot p \,\mathrm{d}x \qquad \forall p \in \hat{V}, \\ \int_\Omega \tfrac{\partial v}{\partial t} \cdot q \,\mathrm{d}x &= \int_{\partial\Omega} bu \cdot q \,\mathrm{d}\boldsymbol{s} - \int_\Omega bu \cdot \tfrac{\partial q}{\partial x} \,\mathrm{d}x + \int_\Omega f \cdot q \,\mathrm{d}x \qquad \forall q \in \hat{U}, \end{aligned} \tag{4.2}$$

where $\mathrm{d}\boldsymbol{s} = \boldsymbol{n}\,\mathrm{d}s$ indicates multiplication with the normal outward pointing unit vector $\boldsymbol{n}$ and integration over facets. On a one dimensional domain of interest $\Omega = [a, b]$, we define $\boldsymbol{n}(a) = 1$ and $\boldsymbol{n}(b) = -1$. The integral over the boundary can be split up over the Dirichlet boundary Γ_D where u is given and the Neumann boundary Γ_N where $u(x) \cdot \boldsymbol{n}$ is given

$$\int_{\partial\Omega} bu \cdot q \,\mathrm{d}\boldsymbol{s} = \int_{\partial\Gamma_D} bu \cdot q \,\mathrm{d}\boldsymbol{s} + \int_{\partial\Gamma_N} bu \cdot q \,\mathrm{d}\boldsymbol{s} \qquad \forall q \in \hat{U}. \tag{4.3}$$

Letting the test functions q vanish on the Dirichlet boundary Γ_D where the solution u is known, and assuming only homogeneous Neumann boundary conditions, i.e., $u(x) \cdot \boldsymbol{n} = 0$ for $x \in \Gamma_N$, we arrive at the following variational problem: find $u \in U$ and $v \in V$ such that

$$\begin{aligned} \int_{\Omega} \tfrac{\partial u}{\partial t} \cdot p \,\mathrm{d}x &= \int_{\Omega} a \tfrac{\partial v}{\partial x} \cdot p \,\mathrm{d}x \qquad \forall p \in \hat{V}, \\ \int_{\Omega} \tfrac{\partial v}{\partial t} \cdot q \,\mathrm{d}x &= -\int_{\Omega} bu \cdot \tfrac{\partial q}{\partial x} \,\mathrm{d}x + \int_{\Omega} f \cdot q \,\mathrm{d}x \qquad \forall q \in \hat{U}, \end{aligned} \tag{4.4}$$

where the test spaces $\hat{U}$ and $\hat{V}$ are defined by

$$\hat{U} = \left\{ u \in H^1(\Omega) : u = 0 \,\mathrm{on}\, \Gamma_D \right\}, \quad \hat{V} = \left\{ v \in H^1(\Omega) \right\}, \tag{4.5}$$

and the trial spaces U and V are defined by

$$U = \left\{ u \in H^1(\Omega) : u = u_0 \,\mathrm{on}\, \Gamma_D \right\}, \quad V = \hat{V}, \tag{4.6}$$

$H^1(\Omega)$ is a first order Sobolev space on the domain of interest Ω, $u(x,t) = u_0(x,t)$ for $x \in \Gamma_D$ is the given Dirichlet boundary conditions.

Discrete variational formulation

Next, the continuous variational problem is approximated with a discrete variational problem posed on finite dimensional subspaces $U_h \subset U$, $V_h \subset V$, $\hat{U}_h \subset \hat{U}$ and $\hat{V}_h \subset \hat{V}$. The discrete variational problem reads: find $u_h \in U_h$ and $v_h \in V_h$ such that

$$\begin{aligned} \int_{\Omega_h} \tfrac{\partial u_h}{\partial t} \cdot p_h \,\mathrm{d}x &= \int_{\Omega_h} a \tfrac{\partial v_h}{\partial x} \cdot p_h \,\mathrm{d}x \qquad \forall p_h \in \hat{V}_h, \\ \int_{\Omega_h} \tfrac{\partial v_h}{\partial t} \cdot q_h \,\mathrm{d}x &= -\int_{\Omega_h} bu_h \cdot \tfrac{\partial q_h}{\partial x} \,\mathrm{d}x + \int_{\Omega_h} f \cdot q_h \,\mathrm{d}x \qquad \forall q_h \in \hat{U}_h, \end{aligned} \tag{4.7}$$

where u_h and v_h are approximations of the unknown functions u and v

$$u \approx u_h = \sum_{i=1}^{N_d} \hat{u}_n^i(t)\phi_i(x), \qquad v \approx v_h = \sum_{i=1}^{N_d} \hat{v}_i(t)\psi_i(x), \tag{4.8}$$

where ϕ_i and ψ_i for $i = 1 \dots N_d$ form a basis for the trial spaces, $\hat{u}_i(t)$ and $\hat{v}_i(t)$ are the degrees of freedom (dofs) and n_d is the number of dofs per unknown function.

Implementation in FEniCS

An implementation of the obtained weak discrete variational formulation in FEniCS/DOLFIN looks like

```
# Initiate test and trial functions
u, v = TrialFunction(U), TrialFunction(V)
p, q = TestFunction(V), TestFunction(U)

# Define Dirichlet boundary conditions
BCu = DirichletBC(U, u0, gamma_E)

# Variational weak formulation as UFL form
F1 = inner(u_dot - a*v.dx(0), p)*dx
F2 = inner(v_dot - f, q)*dx + inner(b*u, q.dx(0))*dx
```

where `U` and `V` are appropriate function spaces, which will be discussed in Section 4.3.1, and `u_dot` and `v_dot` are approximations of the temporal derivatives, which will be introduced in Section 4.4. The function `.dx(i)` indicates a spatial derivative in the ith dimension. The multiplication `*dx` indicates integration over all cells of the mesh. All other symbols have been introduced in the previous section.

The code has to be completed by adding linear solvers, which depend on the chosen time integrators and hence will be added in Section 4.4.

4.2 Finite element formulation of the 3d generic wave equation using discontinuous elements

We use the Discontinuous Galerkin FEM (DG-FEM) for spatial discretisation of the generic wave equation (2.7). As the relation between the generic wave equation and the physical wave problems was given in Section 2.2, we at once obtain a DG-FEM formulation for the acoustic, electromagnetic and elastic wave problems.

DG-FEM first handles the spatial discretisation of every unknown field component q_i for each cell locally. For this, the approximations for q_i are expressed as a summation of local solutions q_i^k:

$$q_i \approx \sum_{k=1}^{N_k} q_i^k, \tag{4.9}$$

where N_k is the number of elements in the mesh. The local solutions q_i^k on cell K_k can be written as

$$x \in K_k : q_i^k(x,t) = \sum_{n=1}^{N_{dk}} \hat{q}_i^{k_n}(t)\boldsymbol{\psi}_n(x), \tag{4.10}$$

where N_{dk} is the number dofs $\hat{q}_i^{k_n}(t)$ per element and the functions ψ_n form a local polynomial basis [51]. Due to the use of cell-wise approximations, multiple approximations exist for the unknown field on element interfaces. E.g., for a point x on the interface between elements K_a and K_b the local solution $q_i^a(x,t)$ can differ from $q_i^b(x,t)$. This effect has to be taken into account when integrating over facets later on.

The DG-FEM discretisation per cell leads to a local variational formulation. Next, the local formulations are summed over all cells of the mesh to obtain a global variational formulation. From now on, we skip the continuous variational formulation, and immediately use finite dimensional function spaces for the test and trial functions to formulate discrete variational formulations. As we no longer use infinite function spaces, we drop the subscript h to indicate the discrete function spaces and their elements.

At the end of the section, we incorporate the boundary conditions into the global variational formulation.

4.2.1 Local variational formulation

To obtain the DG-FEM formulation for the generic wave equation, we multiply (2.7) with a vector of local test functions $\boldsymbol{l} \in \boldsymbol{V}$ and integrate over one mesh cell $K_i \in \Omega_h$.

$$\int_{K_i} \left(\dot{\boldsymbol{q}} + \sum_{i=1}^{3} \boldsymbol{F}_{i,i} - \boldsymbol{f} \right) \cdot \boldsymbol{l} \,\mathrm{d}x = 0. \tag{4.11}$$

After integration by parts, we get the local weak variational formulation

$$\int_{K_i} (\dot{\boldsymbol{q}} - \boldsymbol{f}) \cdot \boldsymbol{l} \,\mathrm{d}x - \int_{K_i} \sum_{i=1}^{3} \boldsymbol{F}_i \cdot \boldsymbol{l}_{,1} \,\mathrm{d}x = - \int_{\partial K_i} \sum_{i=1}^{3} n_i \boldsymbol{F}_i \cdot \boldsymbol{l} \,\mathrm{d}s, \tag{4.12}$$

where $\boldsymbol{n} = (n_1, n_2, n_3)$ is the unit outward pointing normal vector and the notation $\boldsymbol{l}_{,i} = \partial \boldsymbol{l}/\partial x_i$ implies component-wise partial differentiation of $\boldsymbol{l}$ with respect to x_i. Most facets ∂K_i are interior facets; on each such facet two approximations to the vector $\boldsymbol{q}$ exist due to the use of local approximations. We refer to the approximation associated to cell K_i as $\boldsymbol{q}^-$ and the approximation associated to the neighbouring cell that shares the facet as $\boldsymbol{q}^+$. For an exterior facet $\partial K_i \in \partial\Omega$, the approximation $\boldsymbol{q}^+$ does not exist, but can be used for imposing boundary conditions as will be discussed in Section 4.2.3. For numerical stability, a proper combination of both the interior approximation $\boldsymbol{q}^-$ and the exterior approximation $\boldsymbol{q}^+$ has to be used for $\sum_{i=1}^{3} n_i \boldsymbol{F}_i$. We refer to this approximation as the *numerical flux*. We define the numerical fluxes $\boldsymbol{F}^* = \sum_{i=1}^{3} n_i \boldsymbol{F}_i^*$, which leads to the *local variational form*

$$\int_{K_i} (\dot{\boldsymbol{q}} - \boldsymbol{f}) \ \cdot \boldsymbol{l} \, \mathrm{d}x - \int_{K_i} \sum_{i=1}^{3} \boldsymbol{F}_i \ \cdot \boldsymbol{l}_{,1} \, \mathrm{d}x = - \int_{\partial K_i} \boldsymbol{F}^* \ \cdot \boldsymbol{l} \, \mathrm{d}s, \tag{4.13}$$

where we use the Lax-Friedrichs numerical flux

$$\boldsymbol{F}_i{}^* = \{\mathbf{A}_i \boldsymbol{q}\} + \frac{C}{2} n_i^- \, [\boldsymbol{q}], \tag{4.14}$$

and C is the largest wave speed occurring in the wave propagation problem. The interior unit outward pointing normal vector on ∂K_i is denoted as $\boldsymbol{n}^- = \left(n_1^-, n_2^-, n_3^-\right)$[1] For any vector $\boldsymbol{r}$ we define the average and difference as

$$\{\boldsymbol{r}\} = \frac{\boldsymbol{r}^+ + \boldsymbol{r}^-}{2} \quad \text{(average)}, \qquad [\boldsymbol{r}] = \boldsymbol{r}^- - \boldsymbol{r}^+ \quad \text{(difference)}. \tag{4.15}$$

As we consider a conformal mesh that allows material jumps at cell interfaces, every cell can have different $\mathbf{A}_i$ matrices. In that case $\mathbf{A}_i^-$ is the matrix associated with cell K_i and $\mathbf{A}_i^+$ is the matrix associated with the neighbouring cell. In case of a homogeneous material $\mathbf{A}_i^- = \mathbf{A}_i^+$. The Lax-Friedrichs numerical flux guarantees the stability of the scheme [51].

4.2.2 Global variational formulation

To solve the wave equation over the entire domain Ω_h and not only over one cell, we need to obtain a global variational formulation by summing Equation (4.13)

---

[1] In case ∂K_i is an interior facet, the unit outward pointing normal vector of the neighbouring element on the same facet is denoted by $\boldsymbol{n}^+ = \left(n_1^+, n_2^+, n_3^+\right) = -\boldsymbol{n}^-$ and is referred to as the exterior unit outward pointing normal vector on ∂K_i.

over all cells $K_i \in \Omega_h$

$$\sum_{K_i \in \Omega_h} \int_{K_i} (\dot{\boldsymbol{q}} - \boldsymbol{f}) \cdot \boldsymbol{l} \, \mathrm{d}x - \sum_{K_i \in \Omega_h} \int_{K_i} \sum_{i=1}^{3} \boldsymbol{F}_i \cdot \boldsymbol{l}_{,1} \, \mathrm{d}x = - \sum_{K_i \in \Omega_h} \int_{\partial K_i} \boldsymbol{F}^* \cdot \boldsymbol{l} \, \mathrm{d}s. \tag{4.16}$$

The left hand side of (4.16) is easily summed, since there is no overlap between the terms of the different cells. The right hand side of (4.16) contains two contributions for each internal facet, as an internal facet is shared by two cells. We know from [10, eq. (3.3)] that

$$\sum_{K_i \in \Omega_h} \int_{\partial K_i} \boldsymbol{F}^* \cdot \boldsymbol{l} \, \mathrm{d}s = \int_{\Gamma_0} \boldsymbol{F}^* \cdot [\boldsymbol{l}] \, \mathrm{d}s + \int_{\Gamma} \left[\boldsymbol{F}^*\right] \cdot \{\boldsymbol{l}\} \, \mathrm{d}s, \tag{4.17}$$

where $\Gamma = \cup \partial K_i$ is the union of all facets and $\Gamma_0 = \Gamma \backslash \partial \Omega_h$ is the union of all interior facets. Since the numerical flux $\boldsymbol{F}^*$ is uniquely defined on facets, we find that $\left\{\boldsymbol{F}^*\right\} = \boldsymbol{F}^*$ and $\left[\boldsymbol{F}^*\right] = 0$ on interior facets. On exterior facets, all values are uniquely defined, hence we set $\left[\boldsymbol{F}^*\right] = \boldsymbol{F}^*$ and $\{\boldsymbol{l}\} = \boldsymbol{l}$ as in [10]. This leads to the simplification of (4.17)

$$\sum_{K_i \in \Omega_h} \int_{\partial K_i} \boldsymbol{F}^* \cdot \boldsymbol{l} \, \mathrm{d}s = \int_{\Gamma_0} \boldsymbol{F}^* \cdot [\boldsymbol{l}] \, \mathrm{d}s + \int_{\partial \Omega_h} \boldsymbol{F}^* \cdot \boldsymbol{l} \, \mathrm{d}s. \tag{4.18}$$

Using (4.18), we obtain the global weak variational formulation of the generic wave equation (2.7), which reads: find $\boldsymbol{q} \in \boldsymbol{V}$ such that

$$\begin{aligned} \int_{\Omega_h} (\dot{\boldsymbol{q}} - \boldsymbol{f}) \cdot \boldsymbol{l} \, \mathrm{d}x - \int_{\Omega_h} \sum_{i=1}^{3} \boldsymbol{F}_i \cdot \boldsymbol{l}_{,i} \, \mathrm{d}x = \\ - \int_{\Gamma_0} \boldsymbol{F^*} \cdot [\boldsymbol{l}] \, \mathrm{d}s - \int_{\partial \Omega_h} \boldsymbol{F^*} \cdot \boldsymbol{l} \, \mathrm{d}s, \qquad \forall \boldsymbol{l} \in \boldsymbol{V}, \end{aligned} \tag{4.19}$$

where the used function space is

$$\boldsymbol{V} = \left\{ \boldsymbol{v} \in \left[L^2(\Omega)\right]^D : \boldsymbol{v}|_K \in \left[P_k(K)\right]^D \forall K \in \mathcal{T} \right\}, \tag{4.20}$$

where $L^2(\Omega)$ is the function space of all square integrable functions on Ω, $P_k(K)$ is the space of polynomial functions of degree $k \geq 1$ on a cell K and D is the dimension of the vector $\boldsymbol{q}$, which depends on the considered wave propagation problem.

4.2.3 Boundary conditions

In this work, we only consider boundaries where exactly one of both fields occurring in the wave equation is known. This kind of boundaries is incorporated in the DG-FEM formulation by setting fictitious exterior values for $\boldsymbol{q}^+$ on exterior facets [51, Section 7.1.1], [23]. For this, we split up the vector of unknowns $\boldsymbol{q} = (\boldsymbol{q}_1, \boldsymbol{q}_2)$, where $\boldsymbol{q}_1$ contains al components of one unknown field and $\boldsymbol{q}_2$ all components of the other unknown field. When setting the boundary condition $\boldsymbol{q}_1 = \boldsymbol{g}$, the fictitious exterior values are set to

$$\boldsymbol{q}_1^+ = -\boldsymbol{q}_1^- + 2\boldsymbol{g}, \qquad \boldsymbol{q}_2^+ = \boldsymbol{q}_2^-, \tag{4.21}$$

leading to the averages and differences

$$\{\boldsymbol{q}_1\} = \boldsymbol{g}, \qquad \{\boldsymbol{q}_2\} = \boldsymbol{q}_2^-, \qquad [\boldsymbol{q}_1] = 2\boldsymbol{q}_1^- - 2\boldsymbol{g}, \qquad [\boldsymbol{q}_2] = 0. \tag{4.22}$$

When setting the boundary condition $\boldsymbol{q}_2 = \boldsymbol{g}$, it suffices to interchange the indices in the above expressions.

4.2.4 Implementation in FEniCS

An implementation of the obtained weak discrete variational formulation in FEniCS/DOLFIN looks like

```
V = VectorFunctionSpace(mesh, "DG", P, D)

q = TrialFunction(V); l = TestFunction(V)

F1, F2, F3 = A1*q, A2*q, A3*q
F1bnd, F2bnd, F3bnd = A1*(G1*q + g), A2*(G1*q + g), A3*(G1*q + g)

Fstar    =   n[0]('-')*avg(F1) + n[1]('-')*avg(F2)
           + n[2]('-')*avg(F3) + 0.5*C*dif(q)
Fstarbnd = n[0]*F1bnd + n[1]*F2bnd + n[2]*F3bnd + 0.5*C*2*(G2*q+g)

Ft = inner(qdot - f, l)*dx - (inner(F1, l.dx(0))
   + inner(F2, l.dx(1)) + inner(F3, l.dx(2)))*dx
   + inner(Fstar, dif(l))*dS + inner(Fstarbnd, l)*ds

solve(Ft == 0)
```

where the first line will create a vector function space with dimension `D` of discontinuous Lagrange polynomials of order `P` defined on the elements of the considered `mesh`. The matrices `A1`, `A2` and `A3` are the $\mathbf{A}_i$ matrices of the problem. The variable `qdot` is the approximation of the temporal derivative, which will be defined for the different time integrators in Section 4.4. The matrices `G1` and `G2` are used to impose the boundary conditions and look like

$$G1 = \begin{pmatrix} \mathbf{I}_{D_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \qquad G2 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{D_2} \end{pmatrix}, \tag{4.23}$$

where $\mathbf{I}_n$ is the unit matrix of size n, and D_1 is the number of components in the first unknown field and D_2 is the number of components in the second unknown field, and hence $D = D_1 + D_2$. The factor `*dS` indicates integration over all internal facets and `*ds` indicates integration over all exterior facets.

4.3 Finite element formulation of acoustic, electromagnetic and elastic wave equations using conforming elements

Obtaining an FE formulation for the acoustic, electromagnetic and elastic wave propagation problems is done by following the same procedure as used for the one dimensional abstract wave equation. The main point of attention is the used function spaces. As different spatial differential operators appear in the different wave equations, the function spaces containing the trial functions need to be conforming with the applied operators. As the trial functions are multiplied with test functions in the variational forms, also the function spaces of the test functions need to be chosen carefully to make sure these products are well defined.

Choosing appropriate conforming elements is a well documented problem, and will be summarised in Section 4.3.1. In the following sections, discrete variational formulations for the acoustic, electromagnetic and elastic wave problems will be given, together with a code snippet to indicate the implementation in FEniCS/DOLFIN. Special attention will be given to the elastic wave problem, as we opt to impose the symmetry of the stress tensor differently when using FEM compared to the approach when using FIT and DG-FEM.

4.3.1 Conforming elements

Conforming elements are described generically in finite element exterior calculus (FEEC) [11]. Conforming elements are also discussed in the different wave problem fields separately, e.g., *Maxwell's house* for the electromagnetic wave problem in [18]. As FEEC is too general and Maxwell's house is too specific, we opt for a middle way and use the so called Whitney complex as presented in Geometric Integration Theory [100]. The Whitney complex is referred to as a discretised cohomology related to de Rham cohomology, which was generalized in Hodge theory [53], which is one aspect of the study of differential forms on smooth manifolds.

In summary, all we need is the fact that the Sobolev spaces in the definition of the used function spaces have to be conforming to the differential operators that are used on the test and trial functions. A gradient should only be applied to gradient conforming elements. The same holds for divergences and curls. The common H^1 Sobolev space is gradient conforming. Divergence conforming Sobolev spaces are denoted as $H\,(\mathrm{div})$ and curl conforming Sobolev spaces are denoted as $H\,(\mathrm{curl})$. The relevant properties of the Whitney complex state that the space spanned by applying a gradient on all elements of a gradient conforming space is embedded in a curl conforming space. In its turn the space spanned by applying a curl on all elements of a curl conforming space is embedded in a divergence conforming space. And finally, the space spanned by applying a divergence on all elements of a divergence conforming space is embedded in a gradient conforming space. These properties can be summarised by the schematic

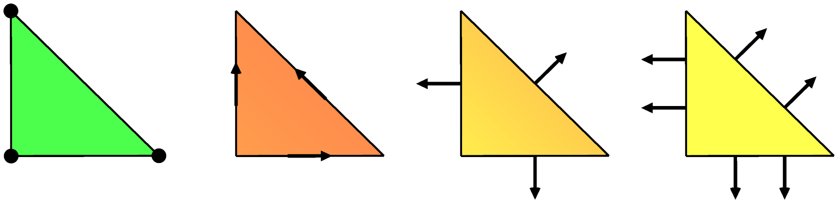
$$H^1 \xrightarrow{\mathbf{grad}} H\,(\mathrm{curl}) \xrightarrow{\mathbf{curl}} H\,(\mathrm{div}) \xrightarrow{\mathbf{div}} H^1. \tag{4.24}$$

4.3.2 Finite element formulation of the acoustic wave equation

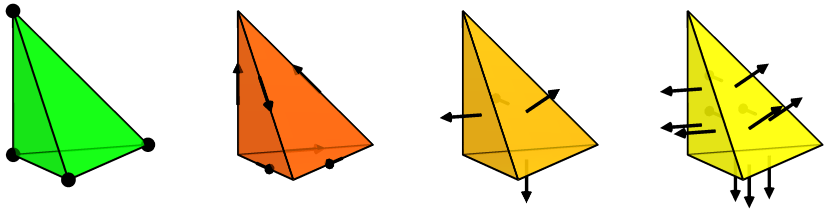
The semi-discrete FE formulation of the acoustic problem (2.8) reads: find $\boldsymbol{v} \in \boldsymbol{U}$ and $p \in W$ such that

$$\begin{aligned} \int_\Omega w \frac{1}{K} \dot{p}\,\mathrm{d}x &= -\int_\Omega w \nabla \cdot \boldsymbol{v}\,\mathrm{d}x \quad \forall w \in W, \\ \int_\Omega \boldsymbol{u} \cdot \rho \dot{\boldsymbol{v}}\,\mathrm{d}x &= \int_\Omega \nabla \cdot \boldsymbol{u} p\,\mathrm{d}x + \int_\Omega \boldsymbol{u} \cdot \boldsymbol{f}\,\mathrm{d}x \quad \forall \boldsymbol{u} \in \boldsymbol{U}, \end{aligned} \tag{4.25}$$

where the function space $W \subset H^1(\Omega)$ is the usual continuous Lagrange finite element space (See Figures 4.1a and 4.1e) and $\boldsymbol{U} \subset H(\mathrm{div}, \Omega)$ is spanned by Raviart–Thomas elements [82] (See Figures 4.1c and 4.1g). We use the same polynomial order for both finite element spaces.



(a) Lagrange. (b) Nédélec. (c) Raviart-Thomas. (d) Brezzi-Douglas-Marini.



(e) Lagrange. (f) Nédélec. (g) Raviart-Thomas. (h) Brezzi-Douglas-Marini.

Figure 4.1: Visual representation of the finite elements of first order, in 2D (top) and 3D (bottom), used for the finite element formulation of the different wave propagation problems described in Section 4.3. Figures copied from [68].

FEniCS implementation

An implementation of the obtained weak discrete variational formulation in FEniCS/DOLFIN looks like

```
U = FunctionSpace(mesh, "RT", P)
W = FunctionSpace(mesh, "CG", P)

w, u = TestFunction(W), TestFunction(U)
p, v = TrialFunction(W), TrialFunction(U)

F1 = inner(w, pdot/K)*dx + inner(w, div(v))*dx
F2 = inner(rho*v-f, u)*dx - inner(div(u), p)*dx
```

where `P` is the polynomial order of the used elements.

4.3.3 Finite element formulation of the electromagnetic wave equation

The semi-discrete finite element formulation for the electromagnetic wave propagation problem (2.10) reads: find $\boldsymbol{h} \in \boldsymbol{U}$ and $\boldsymbol{e} \in \boldsymbol{V}$ such that

$$\begin{aligned} \int_\Omega \boldsymbol{u} \cdot \mu \dot{\boldsymbol{h}} &= - \int_\Omega \boldsymbol{u} \cdot \nabla \times \boldsymbol{e} \, \mathrm{d}x \quad \forall \boldsymbol{u} \in \boldsymbol{U}, \\ \int_\Omega \boldsymbol{v} \cdot \varepsilon \dot{\boldsymbol{e}} &= \int_\Omega \nabla \times \boldsymbol{v} \cdot \boldsymbol{h} \, \mathrm{d}x - \int_\Omega \boldsymbol{v} \cdot \boldsymbol{J} \quad \forall \boldsymbol{v} \in \boldsymbol{V}, \end{aligned} \tag{4.26}$$

where the function spaces $\boldsymbol{U} \subset H(\mathrm{div}, \Omega)$ and $\boldsymbol{V} \subset H(\mathrm{curl}, \Omega)$ are spanned by Raviart–Thomas elements [82] (See Figures 4.1c and 4.1g) and Nédélec elements of the first kind [75] (See Figures 4.1b and 4.1f), respectively. We use the same polynomial order for both kinds of elements.

FEniCS implementation

An implementation of the obtained weak discrete variational formulation in FEniCS/DOLFIN looks like

```
U = FunctionSpace(mesh, "RT", P)
V = FunctionSpace(mesh, "Ncurl1", P)

u, v = TestFunction(U), TestFunction(V)
h, e = TrialFunction(U), TrialFunction(V)

F1 = inner(u, mu*h)*dx + inner(u, curl(e))*dx
F2 = inner(epsilon*e+j, v)*dx - inner(curl(v), h)*dx
```

where `P` is the polynomial order of the used elements.

4.3.4 Finite element formulation of the elastic wave equation

As the stress is a symmetric tensor, it is desired that the numerical approximation to the stress is also symmetric. When using FIT and DG-FEM, we imposed this symmetry in a strong way, by only using one approximation for the couples of shear stresses that are identical. Conforming elements that preserve symmetry in a strong way use a large number of degrees of freedom [4, 9]. As these specific function spaces are not available in FEniCS/DOLFIN, we do not consider this option further. Instead, we opt to weakly impose symmetry in the variational formulation. An overview of the history and available methods is given in [7]. In this work, we will only consider the method proposed in [12].

To weakly impose symmetry to the model, we introduce the rotation field $\boldsymbol{r}$

$$\dot{\boldsymbol{r}} = \frac{\nabla v - (\nabla v)^T}{2}, \tag{4.27}$$

and reformulate (2.12) as

$$\begin{aligned} \boldsymbol{\mathcal{C}^{-1}} : \dot{\boldsymbol{\sigma}} &= \nabla \boldsymbol{v} - \dot{\boldsymbol{r}}, \\ \rho \dot{\boldsymbol{v}} &= \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f}. \end{aligned} \tag{4.28}$$

The finite element variational elastic wave problem with weakly imposed symmetry then reads: find $\boldsymbol{\sigma} \in \boldsymbol{U}$, $\boldsymbol{v} \in \boldsymbol{V}$ and $\boldsymbol{r} \in \boldsymbol{W}$ such that

$$\begin{aligned} \int_\Omega \left(\boldsymbol{\mathcal{C}^{-1}} : \dot{\boldsymbol{\sigma}} + \dot{\boldsymbol{r}}\right) : \boldsymbol{\tau} \,\mathrm{d}x + \int_\Omega \boldsymbol{v} \cdot (\nabla \cdot \boldsymbol{\tau}) \,\mathrm{d}x &= 0 \quad \forall \boldsymbol{\tau} \in \boldsymbol{U}, \\ \int_\Omega (\rho \dot{\boldsymbol{v}} - \nabla \cdot \boldsymbol{\sigma} - \boldsymbol{f}) \cdot \boldsymbol{u} &= 0 \quad \forall \boldsymbol{u} \in \boldsymbol{V}, \\ \int_\Omega \dot{\boldsymbol{\sigma}} : \boldsymbol{w} &= 0 \quad \forall \boldsymbol{w} \in \boldsymbol{W}, \end{aligned} \tag{4.29}$$

where the function space $\boldsymbol{U} \subset H(\mathrm{div}, \Omega)$ is spanned by Brezzi-Douglas-Marini elements (See Figures 4.1d and 4.1h) and the spaces $\boldsymbol{V} \subset L^2(\Omega)$ and $\boldsymbol{W} \subset L^2(\Omega)$ are spanned by the usual discontinuous Lagrange elements (See Figures 4.1a and 4.1e). This combination of finite elements is also known as the Arnold-Falk-Winther elements [7, 12]. The order of the elements spanning $\boldsymbol{V}$ and $\boldsymbol{W}$ is one lower than the order of the elements spanning $\boldsymbol{U}$.

The symmetry of the stress is expressed by the last equation of (4.29) as the test functions q are second order anti-symmetric tensors with constant zero diagonal.

FEniCS implementation

An implementation of the obtained weak discrete variational formulation in FEniCS/DOLFIN looks like

```
U = VectorFunctionSpace(mesh, "BDM", P+1)
```

```
V = VectorFunctionSpace(mesh, "DG", P)
W = FunctionSpace(mesh, "DG", P)

t, u, w = TestFunction(U), TestFunction(V), TestFunction(W)
s, v, r = TrialFunction(U), TrialFunction(V), TrialFunction(W)

# Skew matrix with given axial vector
def skw(r):
  if r.function_space().dim() is 1:
    return as_matrix([[0.0, r], \
                      [-r, 0.0]])
  if r.function_space().dim() is 3:
    return as_matrix([[0, -r[2], r[1]], \
                      [r[2], 0, -r[0]], \
                      [-r[1], r[0], 0]])

F1 = .5/mu * (inner(sigmadot, tau) \
             - lam/(2.*mu+3.*lam)*tr(sigmadot)*tr(tau))*dx \
   + inner(vold,div(tau))*dx + inner(skw(rdot), tau)*dx

F2 = rho*inner(vdot, w)*dx - inner(w, div(sigmaold))*dx \
   - inner(f, w)*dx
F3 = inner(sigmadot, skw(q))*dx
```

where `P` is the polynomial order of the used discontinuous elements.

4.4 Time stepping methods

After obtaining an FE formulation, we have to discretise the time derivatives. In addition to the time integrators used in Chapter 3, we also consider the implicit trapezoidal rule. As FEM requires to solve linear systems regardless of the used time integrator, an implicit time integrator does not add a large computational cost.

In the following subsections we give the UFL formulation for the one dimensional abstract wave equations. The implementation for all other presented FE formulations is analogous.

4.4.1 Classical leapfrog

The leapfrog method as introduced in Section 3.3.1 is implemented in FEniCS/DOLFIN as follows:

```
u_dot, v_dot = (u - u0)/dt,(v - v0)/dt
u_old, v_old = u0, v0

F1 = inner(u_dot - a*v_old.dx(0), p)*dx
F2 = inner(v_dot - f, q)*dx + inner(b*u_old, q.dx(0))*dx

while t < T
  solve(F1 == 0, u)
  u0. assign(u)
  solve(F2 == 0, v)
  v0.assign(v)
```

4.4.2 Trapezoidal rule

The trapezoidal rule is a one step implicit Runge-Kutta method [49] that uses central difference for the time derivatives. As the wave equations are discretised at half time steps, also the solutions need to be averaged out over two time steps to ensure a second order approximation. An implementation of the trapezoidal rule in FEniCS/DOLFIN looks like

```
u_dot, v_do = (u - u0)/dt, (v - v0)/dt
u_old, v_old = (u + u0)/2., (v + v0)/2.

F1 = inner(u_dot - a*v_old.dx(0), p)*dx
F2 = inner(v_dot - f, q)*dx + inner(b*u_old, q.dx(0))*dx

F = F1 + F2

while t < T
  solve(F == 0, q)
  q0.assign(q)
```

where `q` is a vector that contains both `u` and `v` and `q0` is defined analogously.

4.4.3 Composition methods

Composition methods as introduced in Section 3.3.3 are implemented in FEniCS/DOLFIN as follows:

```
u_dot, v_dot = (u - u0)/(0.5*dt), (v - v0)/dt
u_old, v_old = u0, v0

F1 = inner(u_dot - C1*a*v_old.dx(0), p)*dx
F2 = inner(v_dot - f, q)*dx + C2*inner(b*u_old, q.dx(0))*dx

while t < T
  for i in range(0, s)
    C1 = beta[i+1]+alpha[i]
    C2 = beta[i+1]+alpha[i+1]
    solve(F1 == 0, u)
    u0. assign(u)
    solve(F2 == 0, v)
    v0.assign(v)
  solve(F1 == 0, u)
  u0.assign(u)

  t += dt
```

where `alpha` and `beta` are vectors containing the weighting parameters of the selected composition method.

4.5 Selecting the most appropriate method

When solving a specific problem, many choices have to be made to create a complete procedure to obtain a numerical solution. We leave the choice between structured or unstructured grids to the user as it depends on the desired accuracy and geometry of the problem. In this chapter, we focus on FE methods that use unstructured grids, either with continuous, conforming elements or discontinuous elements. The next decisions concern the polynomial order of the finite elements and the time integrator. Which method to use for solving the obtained linear system in every time step generally follows from the previously made choices.

In Section 4.5.1 we discuss and prioritise the features we desire from a solution procedure. This will help us to discuss the pros and cons of the different options that are available for the choices that need to be made in Section 4.5.2. We illustrate the influence of the different options by comparing different combinations for an elastic wave propagation problem in Section 4.6.1.

4.5.1 Desired features of the used method

To make suitable choices to obtain an optimal solution method for a problem, we have to be clear on what properties are desired from the solution. Recall that solving forward wave propagation problems will only be one step in the process to solve inverse problems in the following chapter. To solve an inverse problem, we have a model, a partial parameter set and a set of observations. These observations will have a limited accuracy due to, a.o., noise and measurement error.

Accuracy

Higher accuracy is generally obtained through more computational work, which can be the consequence of a denser mesh, a smaller time step size and higher order approximations of the derivatives in time and space. As we will compare the result of the numerical procedure with measured data, achieving high accuracy is not a high priority. Hence we can sacrifice some accuracy to obtain faster approximation methods. Of course we require at least some correct digits in our solution.

Computational time

Solving inverse problems relies on repeatedly solving forward problems, every time with different parameters. Hence a fast forward solver is a top priority. To obtain a fast inverse solver, it is tempting to choose the option that solves every sub-problem in the solution procedure the fastest. This may however not result in the best computational time for the entire problem, as not all combinations of options go well together.

Memory usage

Low memory usage is desired, as it allows solving larger problems with the same amount of memory. Reducing the memory usage can be done, e.g., by repeating certain computations rather than storing the results, which leads to a prolonged computational time. On the other hand, data locality has become a point of attention due to memory latency, as moving data around has become slow in comparison to executing floating point operations. When little data is needed to obtain the solutions, it can be kept close to the processor, which can speed up the computational time [63].

Scalability

The choices that lead to an optimal procedure for obtaining numerical results sequentially, i.e., on one computer core, may not lead to an optimal procedure for obtaining numerical results in parallel, i.e., on multiple computer cores. As applications can be computationally demanding, it is important to keep an eye on how memory usage and computational time of procedures change when they are being executed in parallel.

4.5.2 Choices that need to be made

All choices that are made to obtain a system of linear equations influences what this linear system will look like and how many systems that have to be solved. In this section, we discuss the pros and cons of the available options. We conclude the section with a discussion of which linear solvers should be used for the obtained linear systems.

Continuous versus discontinuous elements

Continuous elements are more widely used than discontinuous elements. In many cases, the use of continuous elements leads to less dofs to obtain a certain accuracy than the use of discontinuous elements [51]. In forward problems, the solution of the unknown fields only has to be stored for a few time steps. Hence the number of dofs will have limited effect. However, some inverse solvers need to store the solution in every time step. In that case, the number of dofs can have a significant influence on the memory usage.

Discontinuous elements, when used in combination with an explicit time integrator, allow cell-wise updates of dofs. As linear systems for one cell are relatively small, a factorisation of local matrices can easily be computed and stored, significantly reducing the memory requirements. If all elements of a mesh are a transformation of the same simplex, only one local mass matrix needs to be factorised and stored. This small amount of data can be kept local to the processor, speeding up computations. As the data are used quite locally, parallelisation is straightforward.

When continuous elements are used, factorisations of the obtained linear systems are dense, which results in high memory usage. Alternatively, iterative linear solvers can be used, which do not require factorisations. Computing and storing factorisation of large matrices from three dimensional problems becomes impossible already for coarse meshes.

Polynomial order of the finite elements

As discussed in Section 2.4.1, about twenty dofs per wavelength are desired to ensure that numerical solutions are accurate enough. The number of dofs depend on the mesh density and the order of the used elements. Higher order elements lead to smaller, more complicated linear systems. This makes it again hard to predict which option will result in a faster method. Higher order polynomial orders by definition result in higher accuracy, allowing less cells. The spatial discretisation error is however not always reached by low order time integrators as demonstrated in Chapter 3.

Implicit versus explicit time integration

Implicit time integration leads to more complicated linear systems, which can not be factorised with a limited amount of memory and computations, leading to more computational work. However, implicit methods generally allow larger time step sizes (see Section 2.4.1), which reduces the number of linear systems to solve. As continuous elements automatically require system solves, using implicit methods is quite appealing in combination with continuous elements. When using implicit time integration with discontinuous elements, the possibility to factorise the linear system gets lost, hence making it more appealing to use explicit methods.

Accuracy order of the time integration

The influence of the time integrator was studied for the finite integration technique in Section 3.5 where we concluded that higher order time integrators can be preferred over reducing the time step size. In essence, the aim is to choose time step sizes for the different time integrators that result in precision up the spatial discretisation error. The more the temporal and spatial accuracy differ, the more unnecessary computations will be done. The behaviour of the spatial discretisation error changes with the spatial discretisation method and the polynomial order of the elements. Hence different conclusions might be valid for FIT and FEM.

Solving the linear systems

There are multiple techniques that can be harnessed to solve linear systems. Information about the structure of the system is essential to find the most efficient solution technique. When a factorisation can easily be computed and stored it is often preferred over using iterative methods, as direct approaches result in exact solutions. Also, when deciding which iterative linear solver to use, the structure of the system is crucial. The most generic iterative method is GMRES, which is quite robust but converges slowly if no preconditioner is used. When additional information is known, a faster method can be selected. E.g., when the system is symmetric and positive-definite, a conjugate gradient method is often the method of choice [81].

In the case of linear systems originating from FE discretisations, the structure of the matrices is known. Information about the used elements and polynomial order can be used to speed up the linear solves. E.g., it is known that systems occurring in FE formulations where divergence conforming elements are used, result in saddle point problems [34]. Linear solvers that are very problem specific are not standardly available in FEniCS and will not further be considered. From the available iterative solvers in FEniCS, the Bi-CGSTAB method appears to be the most applicable option for all obtained linear systems.

4.6 Elastodynamic test case

In this section, we study the influence of the different options for numerical simulation of wave propagation problems that were presented and summarised in Section 4.5 for a two dimensional elastodynamic example.

4.6.1 Introduction of the test case

This example concerns a smooth displacement field which satisfies homogeneous displacement boundary conditions (particle velocity $\boldsymbol{v} = 0$ on $\partial\Omega$) on a unit square [7, Example 7.1] with homogeneous isotropic material. When setting the mass density $\rho = 1$, the Lamé parameters $\mu = \lambda = 1$ and applying the source function:

$$f = -u - \begin{pmatrix} -4\pi^2 \sin(\pi x) \sin(\pi y) + 2(1-2x)(1-2y) \\ 2\pi^2 \cos(\pi x) \cos(\pi y) - 6\left(x - x^2\right) - 2\left(y - y^2\right) \end{pmatrix} \sin(t), \quad (4.30)$$

the analytical solution for the particle velocity on the computational domain is

$$v = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \sin(\pi x)\sin(\pi y) \\ x(1-x)\,y(1-y) \end{pmatrix} \cos(t)\,. \tag{4.31}$$

The analytical solution for the strain is

$$\epsilon = \begin{pmatrix} \pi c_x \sin(\pi y) & \frac{\pi \sin(\pi x)\cos(\pi y) + (1-2x)\left(y-y^2\right)}{2} \\ \frac{\pi \sin(\pi x)\cos(\pi y) + (1-2x)\left(y-y^2\right)}{2} & \left(x-x^2\right)(1-2y) \end{pmatrix} \sin(t)\,. \tag{4.32}$$

from which we can compute the stress as

$$\sigma = \begin{pmatrix} 2\epsilon_{xx} + \epsilon_{xx} + \epsilon_{yy} & 2\epsilon_{xy} \\ 2\epsilon_{yx} & 2\epsilon_{yy} + \epsilon_{xx} + \epsilon_{yy} \end{pmatrix}. \tag{4.33}$$

We compute a numerical approximation for this problem with first, second and third order continuous and discontinuous elements, once using one computer core and once using sixteen computer cores. We use four different time integrators: the trapezoidal rule (ITR), a second order composition rule (CO2), the classical fourth order explicit Runge-Kutta method (RK4) and a fourth order composition rule (CO4). When increasing the polynomial order of the test and trial functions, more discretisation points are used per element. We keep the distance between discretisation points constant by decreasing the number of elements in the mesh. More specifically, the coarsest mesh we use is divided in 6×6 square cells, each subdivided in 4 triangles, when using first order elements, a mesh of 3×3 crossed square cells with second order elements and a mesh of 2×2 crossed square cells with third order elements. When using sixteen computer cores, we do not consider the coarsest mesh. Every time the mesh is refined, we also halve the time step size. A too large time step would not be able to compute the solution up to the spatial discretisation error, which would prevent reaching higher accuracies.

4.6.2 Results and discussion

Figures 4.2 and 4.3 show the relative error of the computed solution versus the analytical solution on the final time as a function of the required wall clock time when computing with one and sixteen computer cores respectively. The line style indicates the element order, the marker indicates the element type and the line colour indicates the time integrator as defined in the legend in Table 4.1. As these figures contain a lot of information, we make several observations using plots with a subset of the curves shown in Figures 4.2 and 4.3. The legend of all

Line style ⇕ Element order	Marker ⇕ Element type	Line colour ⇕ Time integrator (order)
— 1st order	○ Continuous	ITR - Trapezoidal rule (2)
−− 2nd order	× Discontinuous	CO2 - Composition rule (2)
. . . 3rd order		RK4 - Runge-Kutta (4)
		CO4 - Composition rule (4)

Table 4.1: Legend of Figures 4.4 to 4.3

plots in this section remains the same. The number of used cores is indicated in the captions. Additionally, the timings and errors are summarised in Table 4.2.

Observation 1. (General, continuous vs. discontinuous) Figures 4.2 and 4.3 and Table 4.2 show that using the same mesh, time step size and element order, continuous elements result in a lower error than discontinuous elements for every time integrator. Hence we can say that the spatial discretisation error is smaller for continuous elements than for discontinuous elements.

Observation 2. (ITR, 1 core, element order) Figure 4.4a shows the results obtained with ITR using one computer core. We see that we get higher accuracy with higher order elements for the same computational time. This is not so clear for the coarsest meshes, as only few time steps are needed for these meshes, which leads to a set-up cost that is high in comparison to the time stepping cost.

Observation 3. (ITR, 1 core, continuous vs. discontinuous) For the results in Figure 4.4a, the continuous elements achieve a more accurate result in less time than discontinuous elements with almost every mesh and element order.

The most notable exception is for the finest mesh when using third order elements. In this case, we learn from Table 4.2 that using discontinuous elements is about 50% faster to achieve a similar accuracy. Since ITR is a second order time integrator, halving the time step size is not sufficient to always achieve the spatial discretisation error. As the spatial discretisation error is larger with discontinuous elements (see Observation 1), the used time step size is sufficient to reach the spatial discretisation error when using discontinuous elements. Hence discontinuous elements are slightly more efficient than the continuous elements for third order elements.

Observation 4. (RK4, 1 core, element order, convergence order) Figure 4.5a shows the results obtained with RK4 using one computer core. In this case, the temporal discretisation error is sufficiently small to achieve the spatial discretisation error. Hence, we do not observe the exception described in

Observation 3. In Figure 4.5a we can see first, second and third order convergence slopes in the curves. Also for this time integrator, continuous elements are more efficient than discontinuous elements using one computer core.

Observation 5. (ITR, 1 vs 16 cores, set-up) Figure 4.4b shows the results obtained with ITR using sixteen computer cores. By comparing the timings for one core and sixteen cores in Table 4.2, we see that the set-up cost using sixteen cores is larger for all methods, i.e., it takes more time to compute the results with the coarsest meshes, leading to slower computations on sixteen cores instead of a speed-up.

Observation 6. (RK4, 1 vs 16 cores, set-up) Figure 4.5b shows the results obtained with RK4 using sixteen computer cores. By comparing the timings for one core and sixteen cores in Table 4.2, we see that the set-up cost using sixteen cores is larger only when using continuous elements. For discontinuous elements only one local mass matrix needs to be assembled. This can be done on every core simultaneously, explaining why there is no noticeable difference in set-up cost for discontinuous elements.

Observation 7. (ITR, 1 vs 16 cores, speed-up) When comparing the timings for one core and sixteen cores in Table 4.2 when using ITR, we find that the speed-up for first order elements is about a factor 5. For continuous elements, as the element order increases, the speed-up decreases to a factor 2.5 and 1.8 for second and third order elements respectively. For discontinuous elements, the speed-up however increases as the element order increases to a factor 5.5 and 6 for second and third order elements respectively.

Observation 8. (ERK, 1 vs 16 cores, speed-up) We compare the timings for one core and sixteen cores in Table 4.2 when using RK4. For continuous elements, we find a speed-up factor of 3, 2.5 and 1.8 for first, second and third order elements respectively. For discontinuous elements we find a speed-up factor of 8, 7.7 and 7.5 for first, second and third order elements respectively.

Observation 9. (ITR vs RK4) When comparing the overall efficiency of ITR and RK4 in Figures 4.2 and 4.3, we see that RK4 is slower than ITR.

Observation 10. (3^{rd} order elements, 1 core, all time integrators) Figure 4.6a shows the results for both continuous and discontinuous third order elements obtained with all time integrators using one computer core. The main attention here goes to the performance of the composition rules. We see that using CO4 is pointless, as CO2 is able to achieve the results up to spatial discretisation error by just fulfilling the CFL condition. Hence there is no need to use a higher order method, even though third order elements are used. For low accuracy, ITR is still faster than CO2, but for higher accuracies CO2 becomes more efficient than ITR with both continuous or discontinuous elements.

Observation 11. (3^{rd} order elements, 16 core, all time integrators) Figure 4.6b shows the results for both continuous and discontinuous third order elements obtained with all time integrators using sixteen computer cores. From Table 4.2 we derive that the speed-up of the finest mesh with second and fourth order composition rules is a factor of 5.1 and 6 respectively.

From Observation 2 we conclude that third order elements are preferable to reduce the computational time. Based on Observations 9 and 10 we decide not to consider higher order time integrators further. This limits the available options to CO2 and ITR either with continuous or discontinuous elements. If higher accuracy is desired, discontinuous elements result in faster computations. For lower accuracy ITR is the fastest of the studied methods. When using ITR on one computer core, continuous elements are somewhat faster, but there is very little difference in computational time between continuous and discontinuous elements when using ITR on sixteen computer cores. The CO2 method is only slightly slower than ITR for low accuracies as can be seen in Table 4.2.

Keeping in mind that we are solving forward problems as part of solving inverse problems for which low accuracy is often sufficient, this study suggests that ITR and CO2 are both valuable options.

4.7 Conclusions

In this chapter we presented a finite element formulation for acoustic, electromagnetic and elastic wave propagation problems with continuous and discontinuous elements. When discretising the elastic wave equation, special attention has to be given to imposing symmetry of the discrete approximation of the stress tensor. When using continuous elements, symmetry is imposed weakly, while symmetry is imposed strongly when using discontinuous elements. We studied an example to find a combination of spatial and temporal discretisation techniques to obtain a solution of the wave equations with appropriate accuracy to be used in the inverse problems that will be studied later on in this research. As a result of this study, we will only use continuous and discontinuous finite elements in combination with second order time integrators in the remainder of this research.

	Continuous elements				Discontinuous elements			
	1 core	16 cores	S	accuracy	1 core	16 cores	S	accuracy
ITR $P = 1$	0.51	–		1.75×10^{-2}	0.46	–		1.43×10^{-1}
	1.22	1.62	0.6	4.42×10^{-3}	0.84	1.47	0.6	3.90×10^{-2}
	8.10	2.47	3.3	1.11×10^{-3}	4.32	1.61	2.7	1.02×10^{-2}
	67.13	20.32	3.3	2.77×10^{-4}	33.65	7.51	4.5	2.59×10^{-3}
	812.09	162.88	5.0	6.92×10^{-5}	326.27	65.32	5.0	6.52×10^{-4}
ITR $P = 2$	0.46	–		1.01×10^{-2}	0.42	–		7.08×10^{-2}
	0.87	1.23	0.7	1.30×10^{-3}	0.68	0.97	0.7	1.02×10^{-2}
	4.32	2.36	1.8	1.64×10^{-4}	3.17	1.49	2.1	1.32×10^{-3}
	40.43	16.88	2.4	2.09×10^{-5}	24.36	6.07	4.0	1.70×10^{-4}
	473.04	178.25	2.7	2.80×10^{-6}	244.85	44.95	5.4	2.18×10^{-5}
ITR $P = 3$	0.46	–		5.78×10^{-3}	0.44	–		3.58×10^{-2}
	0.86	1.31	0.7	3.86×10^{-4}	0.71	1.18	0.6	2.59×10^{-3}
	4.10	2.92	1.4	3.02×10^{-5}	3.45	1.59	2.2	1.74×10^{-4}
	39.29	25.59	1.5	4.85×10^{-6}	27.22	6.88	4.0	1.15×10^{-5}
	461.96	259.09	1.8	1.16×10^{-6}	271.03	46.45	5.8	1.25×10^{-6}
RK4 $P = 1$	3.48	–		1.75×10^{-2}	2.00	–		1.43×10^{-1}
	10.78	11.74	0.9	4.42×10^{-3}	6.36	5.045	1.3	4.03×10^{-2}
	57.94	30.80	1.9	1.11×10^{-3}	32.87	12.37	2.6	1.03×10^{-2}
	418.24	203.35	2.1	2.77×10^{-4}	261.39	46.76	5.6	2.58×10^{-3}
	4135.92	1341.13	3.1	6.91×10^{-5}	2225.7	271.13	8.2	6.52×10^{-4}
RK4 $P = 2$	3.44	–		1.01×10^{-2}	1.78	–		7.08×10^{-2}
	8.97	11.34	0.8	1.30×10^{-3}	4.95	4.97	1.0	1.04×10^{-2}
	39.28	30.90	1.3	1.63×10^{-4}	21.85	11.01	2.0	1.34×10^{-3}
	313.62	193.08	1.6	2.04×10^{-5}	165.07	35.40	4.7	1.70×10^{-4}
	3039.84	1212.24	2.5	2.55×10^{-6}	1373.09	179.64	7.6	2.18×10^{-5}
RK4 $P = 3$	3.40	–		5.76×10^{-3}	1.80	–		3.59×10^{-3}
	8.95	12.87	0.7	3.78×10^{-4}	4.97	4.99	1.0	2.69×10^{-3}
	39.91	37.30	1.1	2.40×10^{-5}	22.16	11.55	1.9	1.79×10^{-3}
	334.48	298.96	1.1	1.50×10^{-6}	168.32	37.47	4.5	1.14×10^{-5}
	3195.54	1733.61	1.8	9.40×10^{-8}	1393.33	186.01	7.5	7.22×10^{-7}
CO2 $P = 3$					0.65	–		3.53×10^{-2}
					1.05	1.64	0.6	2.59×10^{-3}
					3.87	2.80	1.4	1.77×10^{-4}
					22.75	7.69	3.0	1.14×10^{-5}
					175.16	34.99	5.0	7.34×10^{-7}
CO4 $P = 3$					1.35	–		3.54×10^{-2}
					3.26	4.49	0.7	2.60×10^{-3}
					12.96	9.17	1.4	1.76×10^{-4}
					78.64	25.95	3.0	1.14×10^{-5}
					611.15	103.35	5.9	7.22×10^{-7}

Table 4.2: Results for the test case presented in Section 4.6.1. The row label indicates the time integrator and element order P, the column label indicates the type of elements. Every cell in the table contains a table with result data. The first column in each cell is the wall clock time used by one computer core to obtain the accuracy in the fourth column. The second column is the wall clock time used by sixteen computer cores to obtain the same accuracy. The third column contains the speedup S going from one to sixteen computer cores.

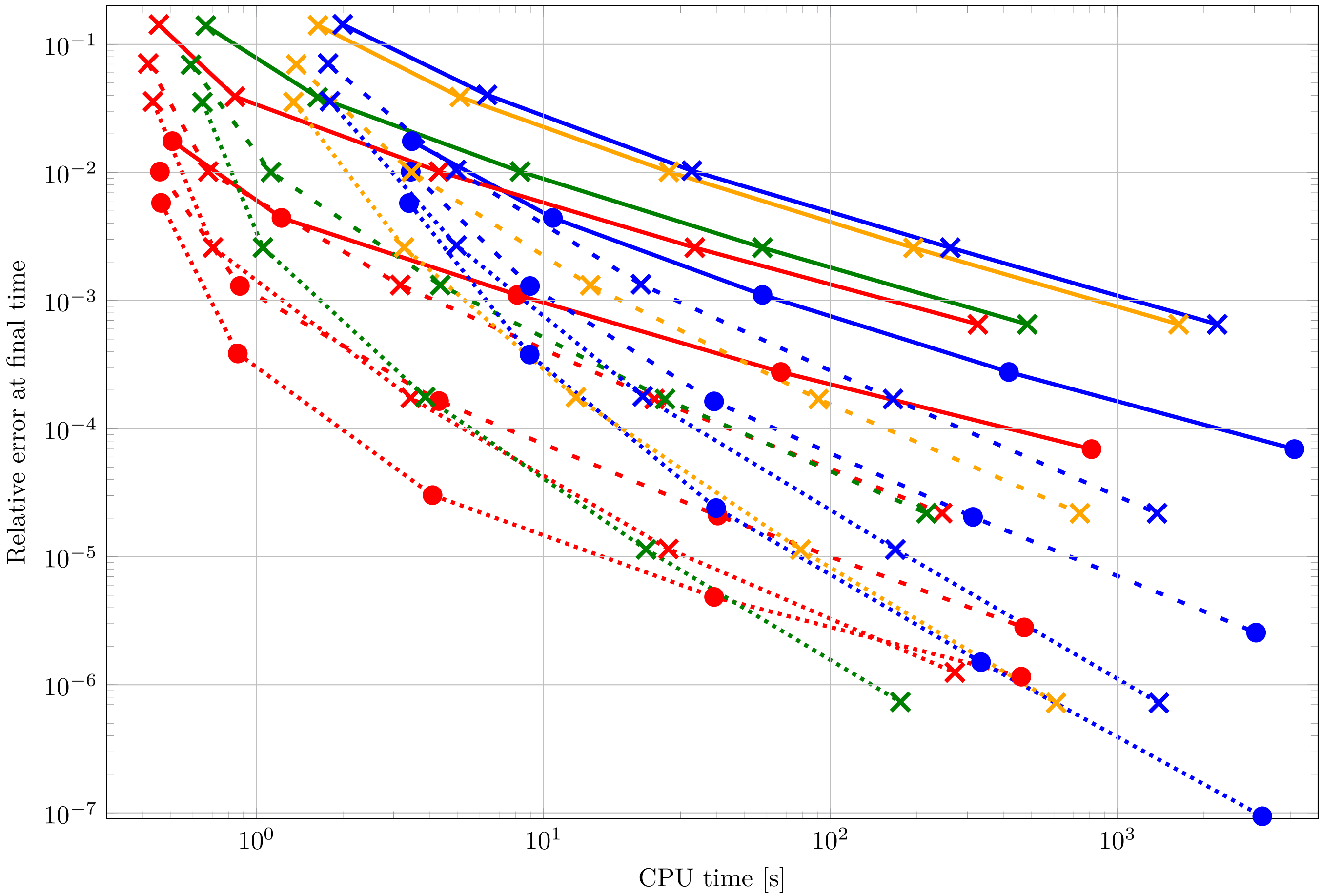


Figure 4.2: Results for the test case presented in Section 4.6.1 obtained with one computer core. The legend can be found in Table 4.1.

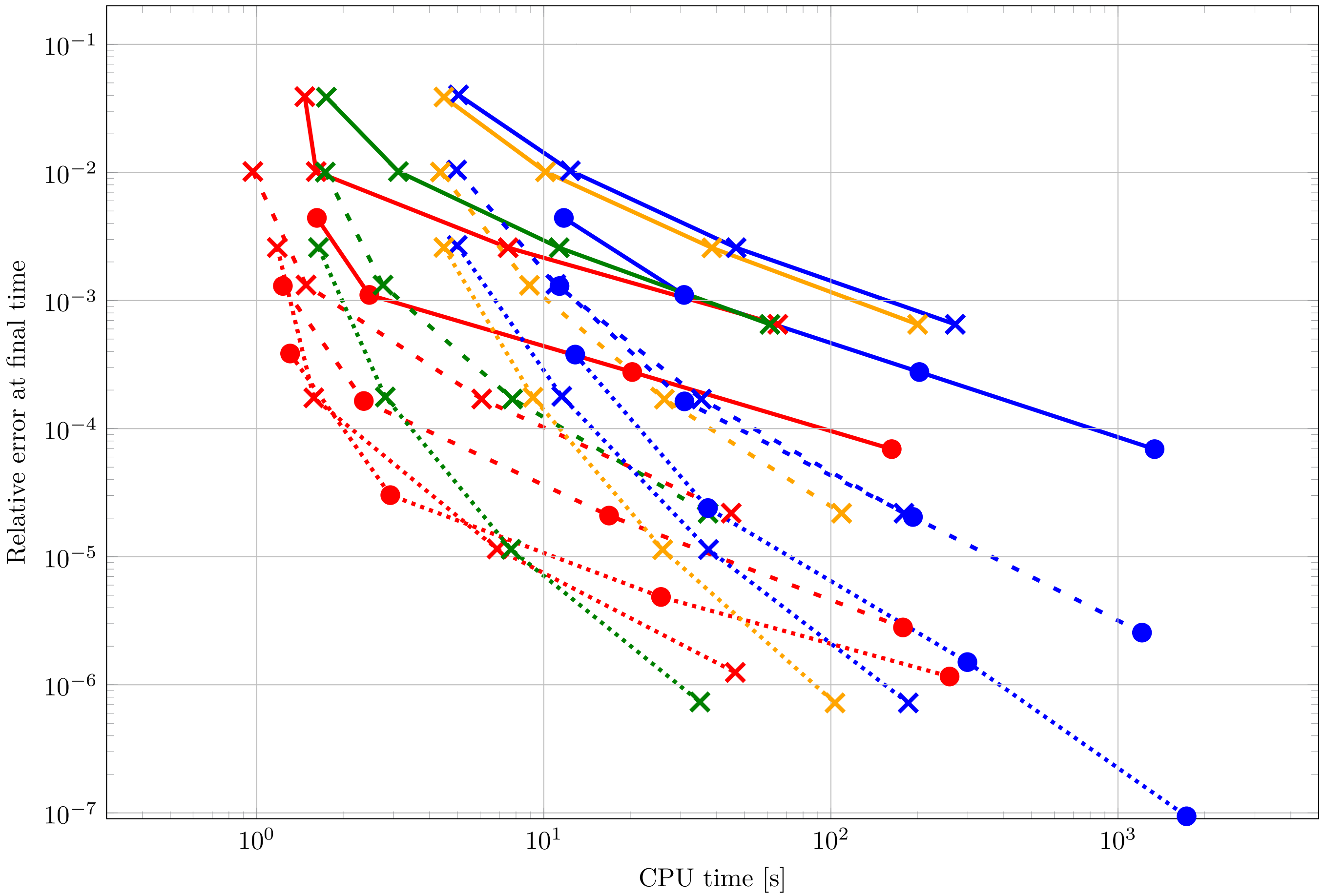
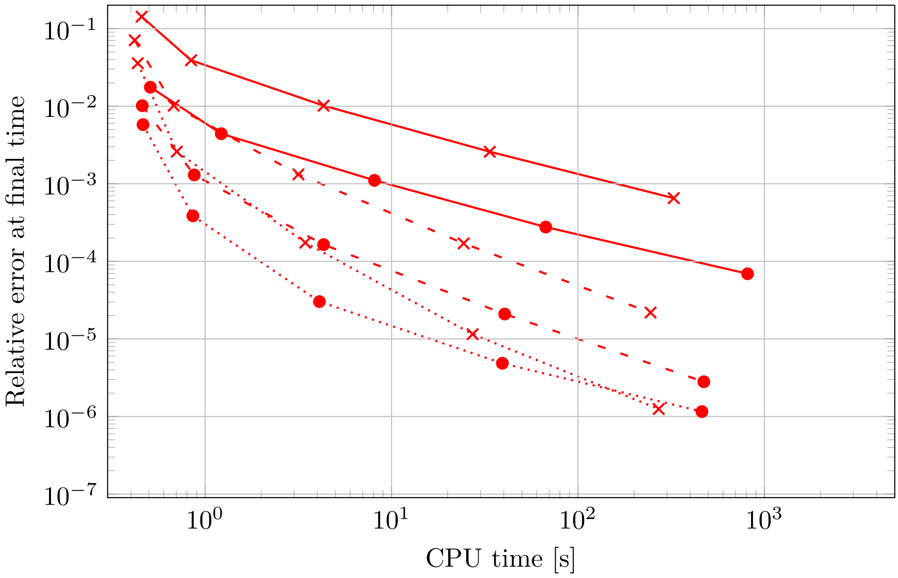
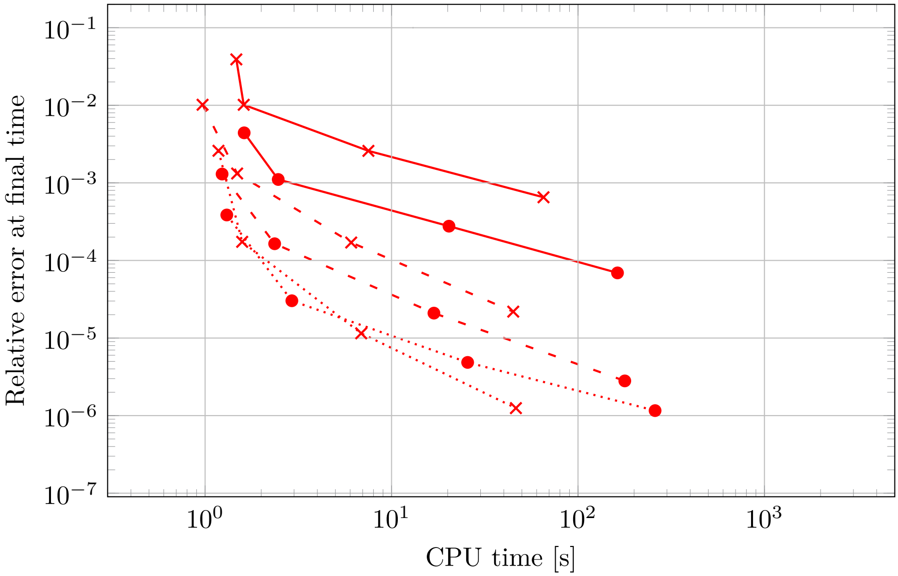


Figure 4.3: Results for the test case presented in Section 4.6.1 obtained with sixteen computer cores. The legend can be found in Table 4.1.

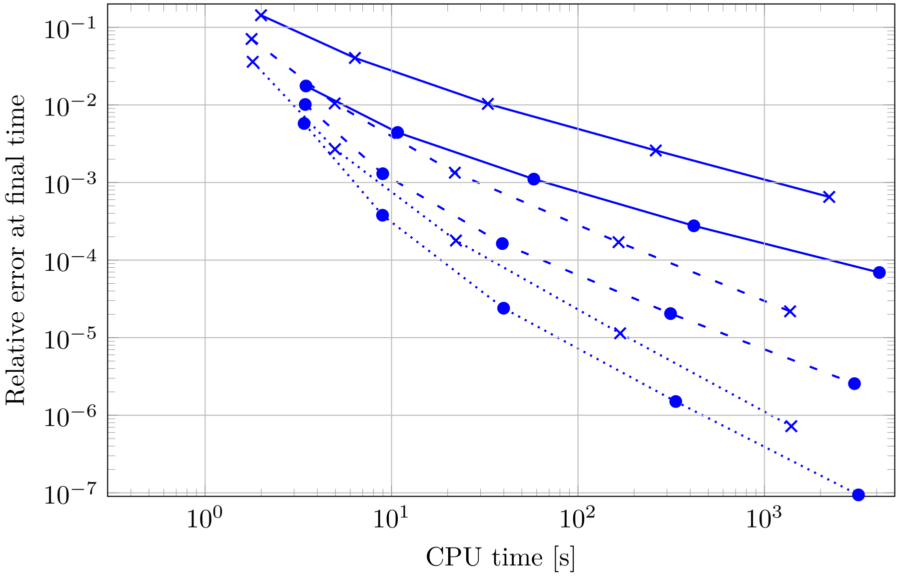


(a) Results obtained with one computer core.

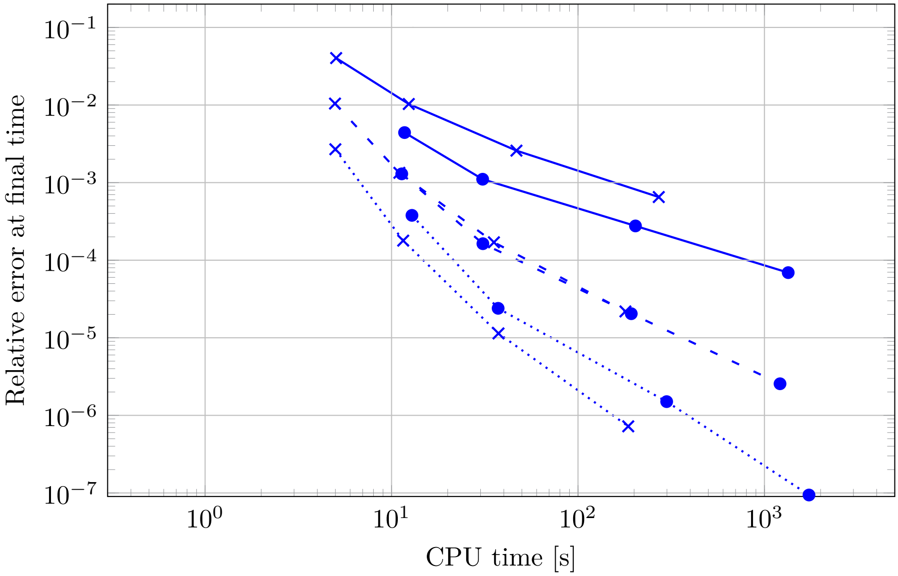


(b) Results obtained with sixteen computer cores.

Figure 4.4: Results for the test case presented in Section 4.6.1 obtained with the trapezoidal rule. The legend can be found in Table 4.1.

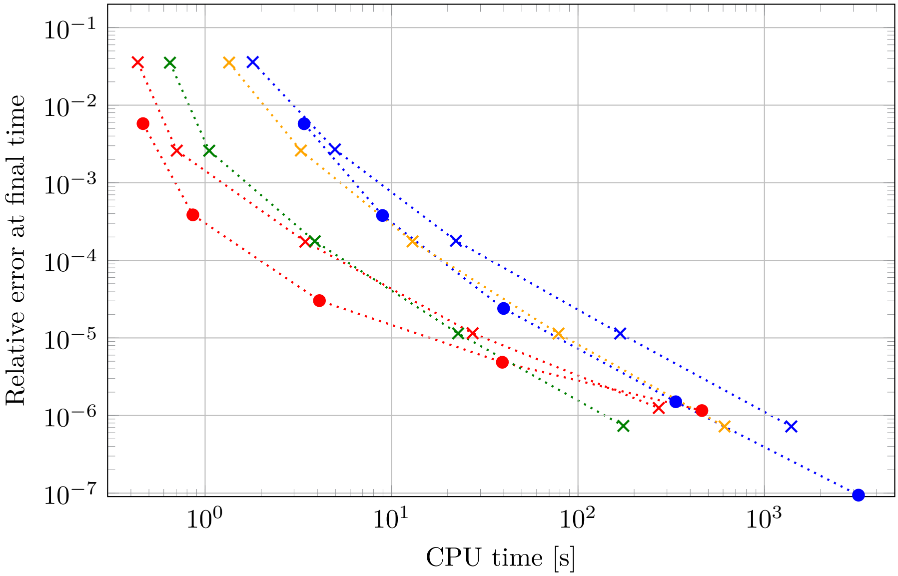


(a) Results obtained with one computer core.

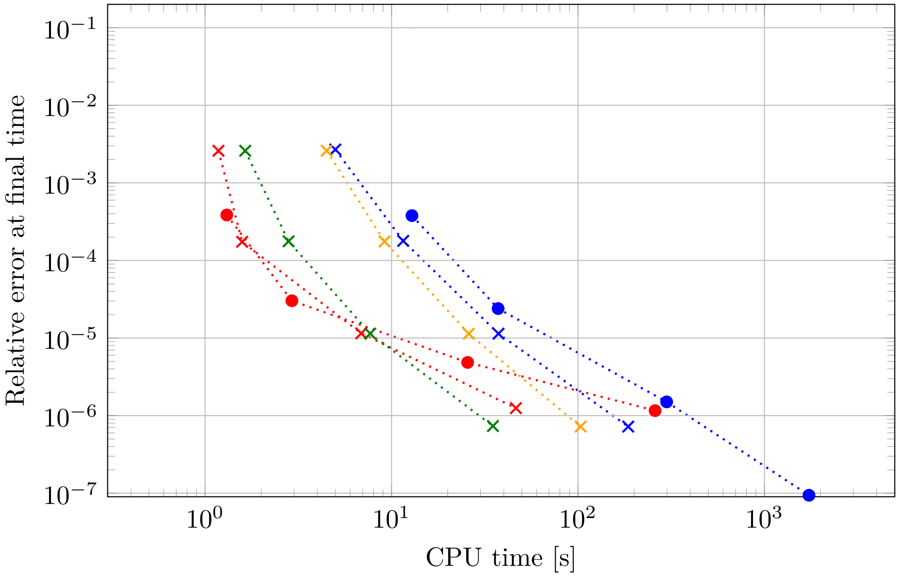


(b) Results obtained with sixteen computer cores.

Figure 4.5: Results for the test case presented in Section 4.6.1 obtained with a fourth order explicit Runge-Kutta time integrator. The legend can be found in Table 4.1.



(a) Results obtained with one computer core.



(b) Results obtained with sixteen computer cores.

Figure 4.6: Results for the test case presented in Section 4.6.1 obtained with third order elements. The legend can be found in Table 4.1.

Chapter 5

Solving inverse problems

The aim of solving inverse problems is to retrieve information on unknown parameters by indirect observations. For inverse wave problems, the unknown parameters of interest can be any of the parameters of the forward problem listed in Section 2.3.1. In NDT&E, several parameters are of interest, leading to a range of inverse problems, as listed in Section 5.1. We give a stochastic and deterministic mathematical formulation of inverse problems in Section 5.2 and point out some of the difficulties involved in solving inverse problems. Sections 5.3 and 5.4 discuss gradient free and gradient based methods respectively, to solve inverse problems. The goal of this chapter is to study gradient based solution methods for inverse problems and to compare problem specific techniques to more general optimisation approaches to solve wave speed determination problems. We first consider constant wave speeds in Section 5.5; This is followed by a discussion on spatially varying wave speeds in Section 5.6.

5.1 Typical inverse problems in non-destructive testing and evaluation

A typical inverse problem in NDT&E is the identification of unknown material parameters. When considering wave propagation problems, most material parameters relate to the wave speeds in the medium. In the simplest cases, e.g., an acoustic wave in a homogeneous fluidum, there is only one wave speed. However, when waves propagate in solids, multiple wave speeds occur. In homogeneous isotropic elastodynamic cases there is only one longitudinal and one transversal wave speed. But in more complicated cases, up to nine spatially dependent wave speeds can occur [3]. Other material parameters, such as attenuation and density, can also be unknown. Here, we will only consider wave speeds. We refer to this kind of problem as a *parameter estimation problem.*

When the material properties are known under normal conditions, we can search for heterogeneities, e.g., damages and defects. Defects come in many shapes and can have different causes. A first kind of defect is the inclusion of impurities in an object, meaning that the material parameters are slightly altered locally, e.g., a block of concrete can have a spatially dependent density due to insufficient mixing. Impurities can be incorporated in the wave model by allowing spatially dependent material parameters. As these inhomogeneities cause scattering of the applied waves, we refer to this kind of problems as *inverse scattering problems* [30].

Another kind of heterogeneity is, e.g., a gas bubble in a cast metal object. We can get a rough idea of where the inclusion is located in the object with an inverse scattering approach, but we no longer have a smoothly varying material parameter as a result of the interface between the gas and the material. Retrieving the shape of included bubbles is a geometry estimation problem rather than a parameter estimation problem. These problems involve an uncertain geometry and hence implicate an uncertain mesh, which requires *shape optimisation* approaches. The latter will not be considered in this work.

5.2 Mathematical formulation of inverse wave propagation problems

Inverse problems start from an input of observed data y_{obs} that depend on the unknown parameters u[1]. The observed data is a combination of the desired information and noise. In this thesis we only assume additive noise, i.e.,

$$y_{\text{obs}} = y(u) + \text{noise}, \tag{5.1}$$

where y is the model that relates the unknown parameters to the observed variables, i.e., the wave equation evaluated at certain time instances and specific spatial coordinates.

The solution of an inverse problem should hold information on how probable any value for an unknown parameter in the admissible set is, which makes statistical inference the most obvious tool. In practice, it is often desired to obtain a likely candidate, or preferably the most likely candidate for the unknown parameters.

---

[1] The variables u and y_{obs} can be scalars, vectors, time and/or space dependent functions, etc.

It may also be desirable to know how likely it is that a selected candidate is actually the correct one.

In the following subsections, we formulate inverse problems both as a stochastic problem and as a deterministic problem. In the remainder of the thesis, we will only solve the deterministic problem, but the stochastic formulation will give us the required insight to properly define a suitable deterministic problem.

5.2.1 Stochastic formulation

The statistical inversion approach is based on the following principles [61, Chapter 3]:

1. All variables in the model are modelled as random variables.
2. The randomness describes the degree of information concerning their realisations.
3. The degree of information concerning these values is coded in the probability distributions.
4. The solution of the inverse problem is the posterior probability distribution.

To properly introduce the stochastic formulation, we reformulate (5.1) in this section as

$$Y = y(U) + E, \tag{5.2}$$

where Y is a random variable representing the observation, U is a random variable representing the unknown parameter and E is a random variable representing the noise. We assume U and E to be mutually independent.

A possible stochastic formulation of the inverse problem reads: How likely is a value for the unknown parameter u, given that we observed y_{obs} under our beliefs about the probability distributions of the unknown parameter and the noise[2]. In other words, we are looking for a posterior probability density $\pi_{\text{post}}\left(u|y_{\text{obs}}\right)$. From Bayesian theory [61, Eq. (3.3)], we know that $\pi_{\text{post}}\left(u|y_{\text{obs}}\right)$ is proportional to the likelihood $\pi_{\text{like}}\left(y_{\text{obs}}|u\right)$ of the observation of y_{obs} when u is the real value for the studied unknown parameter, multiplied with the prior probability density of the unknown parameter $\pi_{\text{prior}}\left(u\right)$:

$$\pi_{\text{post}}\left(u|y_{\text{obs}}\right) \propto \pi_{\text{like}}\left(y_{\text{obs}}|u\right)\pi_{\text{prior}}\left(u\right). \tag{5.3}$$

---

[2] Note the difference in meaning between upper and lower case symbols in this section. Upper case symbols indicate a random variable, while lower case symbols indicate a specific value drawn from the distribution of a random variable.

In the case of additive noise [61, Section 3.2.1], Y conditioned with $U = u$, is distributed as the noise translated with $y(u)$, which results in the likelihood function

$$\pi_{\text{like}}\left(y_{\text{obs}}|u\right) = \pi_{\text{noise}}\left(y_{\text{obs}} - y(u)\right). \tag{5.4}$$

In this work, we assume both the noise and the studied unknown parameter to have a Gaussian distribution, i.e., $E \sim \mathcal{N}\left(0, \Gamma_{\text{noise}}\right)$ and $U \sim \mathcal{N}\left(u_0, \Gamma_{\text{prior}}\right)$, where $\mathcal{N}\left(n_0, \Gamma\right)$ is a normal distribution with mean n_0 and covariance matrix Γ. This brings us to a posterior distribution function of the form

$$\pi_{\text{post}}\left(u|y_{\text{obs}}\right) \propto \exp\left(-\frac{1}{2}\left\|y_{\text{obs}} - y(u)\right\|^2_{\Gamma^{-1}_{\text{noise}}}\right) \exp\left(-\frac{1}{2}\left\|u - u_0\right\|^2_{\Gamma^{-1}_{\text{prior}}}\right), \tag{5.5}$$

where the definition of the norms follow from the spaces to which the observation and unknown parameter belong respectively.

Solving stochastic inverse problems directly

Solving stochastic inverse problems results in more information than solving the deterministic formulation that will be discussed in the following section. In particular, the deterministic approach will result in the most likely candidate for the unknown values, whereas the stochastic approach will result in a probability density function, from which we can extract the likelihood of any possible solution. To obtain this additional information, solving the stochastic inverse problem directly involves a lot more computational work.

One approach to determine the posterior probability density function is to employ sampling techniques. Among the most generic tools for sampling are Markov Chain Monte Carlo methods. In particular the class of Metropolis-Hastings algorithms is often used, as it avoids sampling the involved probability distributions directly, which is usually costly. Markov Chain Monte Carlo methods are especially useful for treating problems with many unknown parameters. As we will not solve stochastic inverse problems directly, we will not discuss this method further. A short discussion of Markov Chain Monte Carlo methods is given in [88] together with an extensive reference list.

5.2.2 Deterministic formulation

The deterministic formulation can be derived from the stochastic formulation in the previous section, by looking for the most likely value for the unknown parameter. Hence we are looking for the maximizer of the posterior probability density function, this is equivalent to finding the minimiser of

$$-\log\Big(\pi_{\text{post}}\left(u|y_{\text{obs}}\right)\Big)\,. \tag{5.6}$$

When we assume the posterior probability density function (5.5) with the covariances $\Gamma_{\text{prior}} = \sigma^2\mathbf{I}$ and $\Gamma_{\text{noise}} = \gamma^2\mathbf{I}$ where $\mathbf{I}$ is the unit matrix and σ and γ are the standard deviation of the prior and the noise probability density function respectively, we are looking to compute the minimiser of

$$\frac{1}{2\sigma^2}\left\|y_{\text{obs}} - y(u)\right\|^2 + \frac{1}{2\gamma^2}\left\|u - u_0\right\|^2, \tag{5.7}$$

where we refer to the first term as the *misfit* between the observed and the computed states, and refer to the second term as the *regularisation* term. Note that the misfit term stems from the uncertainty with respect to the noise and the regularisation from the uncertainty with respect to the unknown parameter.

This inverse problem can be formulated as an abstract optimisation problem with PDE constraints:

$$\min_{y,u} J\left(y\left(u\right), u, y_{\text{obs}}\right) \qquad \text{(objective functional)},$$

$$\boldsymbol{c}(y\left(u\right), u) = 0 \qquad \text{(constraint)},$$

where J is a scalar function referred to as the *objective functional*, y is referred to as the *state* variable, $u \in U_{\text{ad}}$ is called the control variable, U_{ad} is the set of the admissible values for the control variable and $\boldsymbol{c}$ represents a set of constraints. For the considered wave propagation problems, the state variable y contains the computed field $\boldsymbol{q}$ in the wave equation (2.7), and the control variable u contains the studied unknown parameters. Satisfaction of the constraint $\boldsymbol{c}\left(y\left(u\right), u\right)$ in our case is the satisfaction of the wave equation. The objective functional will be a combination of a misfit and a regularisation term as in (5.7).

PDE constrained optimisation problems can be solved with or without using derivatives of the objective and constraints of the optimisation problem. Both approaches are discussed in the subsequent sections. In the following subsection, we first discuss why inverse problems are often hard to solve.

5.2.3 Well-posedness

A problem is well-posed in the sense of Hadamard [47] if (1) a solution exists, (2) the solution is unique and (3) the problem is stable, i.e., the output reacts in a predictable, smooth way to small changes to the input. Forward problems generally fulfil all three requirements and are hence well-posed.

For inverse problems well-posedness is less obvious. Much depends on the exact formulation of the inverse problem. In the case of a stochastic formulation, it does not make much sense to talk about, e.g., uniqueness of the solution. We are mainly interested in the well-posedness for the deterministic formulation presented in Section 5.2.2, as this is the formulation we will solve. We start with the most naive problem formulation: find the value for the unknown parameter u, given measured output y_{obs} and an exact model y, such that

$$y(u) = y_{\text{obs}}. \tag{5.8}$$

Since y_{obs} is influenced by noise, which is not included in the model y, and $y\left(u\right)$ is subject to computational errors, existence of a solution in the admissible set is unlikely, making this naive formulation ill-posed.

Existence of a solution can be solved by lowering the goal of finding a solution for (5.8) to finding a minimiser of a misfit objective

$$\left\|y\left(u\right) - y_{\text{obs}}\right\|. \tag{5.9}$$

Under fairly mild conditions a minimiser is guaranteed to exist. However, uniqueness and stability remain an issue.[3] First assume that the minimiser is not unique, i.e., there exist at least two values for the control u, say u_1 and u_2, that are minimisers of (5.9) where $u_1 \neq u_2$. Then we can add a bias u_0 to the objective with respect to the control, e.g.,

$$\left\|y\left(u\right) - y_{\text{obs}}\right\| + \alpha\left\|u - u_0\right\|. \tag{5.10}$$

The second term, i.e. the bias term, in (5.10) favours the solution of (5.9) that is *closest* to u_0. The parameter α will be discussed at the end of this section. Determining a proper bias will not be discussed in this work.

Even if a unique solution of the inverse problem exists, the inverse problem can still be ill-posed due to instability. An inverse problem is called *unstable* if multiple significantly differing[4] controls u_i exist, such that $y\left(u_i\right) \approx y_{\text{obs}}$ for all i. This problem can again be solved by adding a bias to the objective, which will favour one of the controls u_i.

Adding a bias term regularises the ill-posed problem and results in a well-posed problem. Hence the bias term is also referred to as the regularisation term. The proposed regularisation in (5.10) is also known as *Tikhonov regularisation* [92]. Other kinds of regularisation terms exist, but will not be considered in this work.

---

[3] We give an intuitive explanation why uniqueness and stability are not guaranteed with a misfit objective for the problem of wave speed determination in Section 5.5.1.

[4] If the different controls would be close to each other, all of them would be fairly good approximations meaning that the problem could be considered stable.

Adding a regularisation term to the misfit comes at a cost. Due to the bias term, we are no longer looking for the best candidate to minimise the misfit. Hence, the regularisation parameter α has to be chosen carefully. If α is chosen too large, the misfit will have very little influence, resulting in a useless minimiser. If α is chosen too small, it can be ignored, which may not solve the ill-posedness of the misfit problem.

To determine the regularisation parameter α, remember that we have obtained an objective (5.7) consisting of a misfit term and a regularisation term while searching for the maximum of the posterior probability density function (5.5). In (5.7), the misfit and regularisation term are weighted automatically by the variance that is assumed on the distribution of the noise and the unknown parameter. This formulation provides a suggestion on how to choose the regularisation parameter. We will use this suggestion to set the regularisation parameter. There are however many other strategies to find an optimal regularisation parameter. One is the *Mozorov discrepancy principle* [61, Section 2.3], which roughly states that the regularisation should not try to satisfy the misfit more accurately than up to the noise level.

5.2.4 Inverse crime

When designing solution procedures for inverse problems, it is common to use a simulated reference y_{sim} instead of using a reference y_{obs} that was observed in the real world. When we use a simulated reference $y_{\text{sim}} = y(u)$, where $u \in U_{\text{ad}}$, the inverse problem is less ill-posed than when using an observed reference y_{obs} for two reasons. First, the simulated reference is not affected by noise, while a real observation is (as noted in (5.1)). And second, computing a solution for $y(u)$ is done with a numerical method relying on a mesh which induces numerical errors. As an observed reference is independent of a numerical method and discretisation, the inverse solution method should perform similarly well for any mesh. When using the same numerical method and mesh for solving the inverse problem and for computing the simulated reference, exactly the same numerical errors are included in both results, while this would not be the case when using a different discretisation for both computations.

Replacing the original problem by a less ill-posed problem due to unrealistic assumptions is referred to as committing an *inverse crime* [61, Section 1.2]. Committing the first mentioned inverse crime guarantees existence of a solution, which is unlikely to be the case when using observed data. This first inverse crime can be avoided by adding noise to the computed observation before using it as a reference solution in an inverse problem. The second inverse crime can be avoided by using a different discretisation for computing a simulated reference then for solving the inverse problem.

In this work, both adaptations are carried out in order to approach a relevant situation as close as possible.

5.3 Gradient free solution methods

Gradient free solution methods to solve optimisation problems only use evaluations of the objective and the constraints or rely on an auxiliary forward problem to retrieve likely candidates for the unknown parameters. The gradient is generally unavailable and expensive to obtain, making gradient free optimisation techniques the first option to consider as solution strategy.

5.3.1 Sampling the parameter space

A first method to retrieve the value of an unknown parameter in an inverse problem is to solve the forward problem with every possible value for the unknown parameter and check which outcome matches the observations best. In many problems there is an infinite amount of possible values for the unknown parameter. In such cases, a finite number of possible candidates has to be selected or sampled.

The most general way to obtain a finite number of candidates for an unknown parameter is by trial and error, i.e., randomly select admissible values. The chance of finding the right parameter value increases with every additional sample. The probability of finding the correct value for the unknown parameter by testing a finite number of values in an infinite set is near zero. Nevertheless, retrieving a crude approximation with a trial and error approach is generally feasible in an acceptable amount of time. In order to find a more accurate approximation however, a more sophisticated strategy should be employed to pick subsequent tries. A first structured strategy for sampling is equidistant sampling. In case of a smoothly varying objective, this will result in a most likely value and a bound on the error of this estimate related to the distance between two samples.

However crude and slow, random and equidistant sampling are an option as long as we are looking for one or two scalar parameters. When trying to retrieve more scalar parameters, the size of the parameter space rapidly increases and suffers from the curse of dimensionality. Since the parameters are generally depending on each other, they have to be retrieved simultaneously, which means we are not looking for multiple values each in their own interval, but for a vector of parameters in a multi-dimensional space, which rapidly becomes infeasible for a not too large number of parameters. As we are aiming to solve space dependent parameters, which introduce an unknown scalar parameter per dof, we are facing problems with a large number of parameters. Hence we need to look for more advanced techniques.

5.3.2 More advanced gradient free methods

There exist more advanced gradient free methods than sampling the parameter space to find a minimiser of an objective functional. Giving an exhaustive overview or a detailed explanation of these methods is beyond the scope of this work. We mention only Powell's method and Nelder-Mead, which are methods to find local minima.

Powell's method [79] uses a set of n search vectors. In every iteration, one of the search vectors is replaced. The new search vector is determined by combining the information of the minima along each search vector. The search vector which contributed most in the new search vector is excluded from the set of search vectors. The initial search vectors are typically the normals aligned to each axis.

The Nelder-Mead or Simplex Method [76] uses $n+1$ points in an n-dimensional parameter space to form a simplex. In every iteration, the point with the highest value for the objective functional is replaced by a point with a lower value. Different strategies exist to find a new point. An often used approach is to consider the point obtained by reflecting the point that has to be replaced through the centre of the remaining n points.

These methods are expected to work well when the number of controls is low enough. However, when the dimension of the control space increases, which is the case for spatially dependent controls, the set-up cost of these methods is large. For Nelder-Mead, the objective has to be evaluated in $n+1$ control points, meaning that $n+1$ forward solutions need to be computed. Even more computations need to be done for Powell's method to find minimisers in n directions.

5.3.3 Time reversal

In time reversal approaches [40], a received signal is reversed in time, i.e., t becomes $-t$, and is sent back either from the receiver position (standard time reversal) or from the transmitter position (reciprocal time reversal [25, 33]).

Due to the reciprocity in time of the wave equation, every step in time can be inverted to compute the previous state based on the current state. Doing so, it has been shown that due to the properties of wave problems, it suffices to record the states at several positions over the considered time interval to reconstruct the source signal [41].

Time reversal can be used as part of hybrid methods [59, 66, 80], where we have a measured solution y_{obs} and a computed reference for the same geometry without heterogeneities y_{ref}. The source that would produce the difference $y_{\mathrm{dif}} = y_{\mathrm{obs}} - y_{\mathrm{ref}}$ will be the result of a scatterer. Hence, when $y_{\mathrm{dif}}(-t)$ is sent back from the receiver, there will be a focus at the location of the largest heterogeneity. By using a deflation strategy, also smaller heterogeneities can be retrieved iteratively. In this case, we have reformulated the inverse scattering problem as a *source reconstruction problem*.

The number of transmitters and receivers used for a time reversal procedure is of importance for the quality of the refocused signal. If more receivers are used, more information is captured, and hence the reconstruction will be better. It is possible to compensate for a low number of receivers and transmitters by considering a longer time interval. Each time a wave is reflected by a wall, it can be considered to be a new virtual source. This approach only works in closed systems, as energy is dissipated in (semi-)open systems, causing very little information to be recorded after little time. In media with low attenuation and sufficiently reflecting borders, it can suffice to have only one transmitter and receiver [22, 33, 42]. Changing the number of transmitters and receivers is easily done in computations, but may take quite some effort for experimental set-ups. Using a longer time interval will be no problem for the measurement set-up, but will significantly increase the computational time needed to solve the inverse problem. Because of these opposing interests, time reversal methods will be demanding one way or another.

5.4 Gradient based solution methods

In this section we explain how gradient information can be used to solve PDE constrained optimisation problems. We also indicate how the gradient can be computed for the considered optimisation problems. Only the finite dimensional case, i.e., scalar and vectorial controls, will be considered in this section. We comment on the infinite dimensional case, i.e., control functions, in Section 5.6.3.

5.4.1 Newton type optimisation

For a parameter vector $\boldsymbol{x}^*$ to be an unconstrained minimiser of a vector function $\boldsymbol{f}$, it has to fulfil the optimality condition

$$\nabla \boldsymbol{f}\left(\boldsymbol{x}^*\right) = 0, \tag{5.11}$$

where $\nabla \boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is the gradient of $\boldsymbol{f}$, n is the number of controls in the problem and m is the number of dofs in the forward simulation. The minimisation problem can hence be reformulated as the root finding problem (5.11).

Newton's method is one of the best known root finding algorithms that can also be used for vectorial functions. The idea of Newton's method is to linearise the problem at a position $\boldsymbol{x}_k$, and use the linearised problem to update the root approximation $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$, where $\boldsymbol{p}_k$ is called the *Newton-update.* As linearised model, we use the first order Taylor expansion of $\boldsymbol{f}$ in $\boldsymbol{x}_k$:

$$\nabla \boldsymbol{f}\left(\boldsymbol{x}^*\right) = 0 \approx \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right) \boldsymbol{p}_k. \tag{5.12}$$

From (5.12), we can find the Newton-update

$$\boldsymbol{p}_k = -\left(\nabla^2 \boldsymbol{f}\left(\boldsymbol{x}_k\right)\right)^{-1} \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right), \tag{5.13}$$

where $\nabla^2 \boldsymbol{f}$ is the Hessian of $\boldsymbol{f}$. For this method, the gradient and the Hessian need to be available or computable. Computing the Hessian is not straightforward but possible. In this work, we only consider methods that do not require the Hessian. In particular, we will use Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms, which are *quasi-Newton* methods that use an approximation $\mathbf{B}_k$ to the exact Hessian.

Given $\mathbf{B}_0$ and $\boldsymbol{x}_0$, the BFGS algorithm follows the recipe

1. Determine a *descent direction* $\boldsymbol{p}_k$ by solving the system $\mathbf{B}_k \boldsymbol{p}_k = -\nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right)$.
2. Use a *line search*[5] to find a step size α_k in the direction $\boldsymbol{p}_k$.
3. Compute the new root approximation $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$.
4. Compute the auxiliary variables, containing the update of the root $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \alpha_k \boldsymbol{p}_k$, and of the gradient $\boldsymbol{y}_k = \nabla \boldsymbol{f}\left(\boldsymbol{x}_{k+1}\right) - \nabla \boldsymbol{f}\left(\boldsymbol{x}_k\right)$.
5. Update the Hessian approximation $\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \mathbf{B}_k}{\boldsymbol{s}_k^T \mathbf{B}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k}$.

---

[5] There are many line search strategies available. We use the default option of the used BFGS implementations, which enforce the strong Wolfe conditions [77].

It can be shown that the Hessian approximations fulfil the *secant condition*

$$\mathbf{B}_{k+1}\left(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right)=\nabla\boldsymbol{f}\left(\boldsymbol{x}_{k+1}\right)-\nabla\boldsymbol{f}\left(\boldsymbol{x}_k\right), \tag{5.14}$$

and that $\mathbf{B}_{k+1}$ converges to $\nabla^2\boldsymbol{f}\left(\boldsymbol{x}^*\right)$. The identity matrix can be used as initial Hessian approximation $\mathbf{B}_0$.

A BFGS step requires relatively few additional floating point operations in comparison to the computation of the gradient.

5.4.2 Gradient computation

Computing the derivative of the objective functional $\boldsymbol{J}\big(\boldsymbol{y}(\boldsymbol{u}),\boldsymbol{u}\big)$ and the constraint $\boldsymbol{c}\big(\boldsymbol{y}(\boldsymbol{u}),\boldsymbol{u}\big)=0$ with respect to the vector of control parameters $\boldsymbol{u}$ can be done using the implicit function theorem as

$$\frac{\mathrm{d}\boldsymbol{J}}{\mathrm{d}\boldsymbol{u}}=\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}+\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{u}}, \tag{5.15}$$

and

$$\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}\boldsymbol{u}}=\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}+\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{u}}=0. \tag{5.16}$$

All partial derivatives in (5.15) and (5.16) can be derived analytically. The term $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}$ however is not described by a straightforward equation and hence has to be obtained by solving the tangent linear system

$$\left(\tfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{y}}\right)\left(\tfrac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}\right)=-\left(\tfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{u}}\right). \tag{5.17}$$

The resulting term $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}$ can then be plugged into (5.15) to obtain the derivative of the objective functional with respect to the studied parameter. While the value of $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{u}}$ is only used as an intermediate value, it is often wise to substitute (5.17) in (5.15) resulting in

$$\frac{\mathrm{d}\boldsymbol{J}}{\mathrm{d}\boldsymbol{u}}=-\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{y}}\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{y}}^{-1}\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{u}}+\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{u}}. \tag{5.18}$$

Equation (5.18) can be computed by first solving the *adjoint system*

$$\left(\tfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{y}}\right)^T\boldsymbol{\lambda}=-\left(\tfrac{\partial\boldsymbol{J}}{\partial\boldsymbol{y}}\right)^T \tag{5.19}$$

for the *adjoint state* $\boldsymbol{\lambda}^T$, i.e.,

$$\boldsymbol{\lambda}^T=-\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{y}}\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{y}}^{-1}. \tag{5.20}$$

This allows to compute the derivative of the objective as

$$\frac{\mathrm{d}\boldsymbol{J}}{\mathrm{d}\boldsymbol{u}} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}} + \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{u}}. \tag{5.21}$$

Computing the derivative using the tangent linear system requires solving a linear system for every control parameter, while using the adjoint system only requires one system solve, regardless of the number of parameters. Hence, the adjoint approach is the preferred method for this work.

In this work, we use the dolfin-adjoint library [38, 44], which can automatically provide the gradient for a forward problem implemented with DOLFIN/FEniCS [6, 67, 68]. We will include a code snippet in Section 5.5.7 to indicate how little work it is to set up an optimisation procedure with dolfin-adjoint once a FEniCS implementation for the forward problem is available.

5.5 Wave speed determination

We now consider the problem of wave speed determination. We determine the wave speeds with which elastic waves propagate through a specific object. For this, the object is excited by a known source and only the normal particle velocity at one location on the surface of the object is recorded. We solve an optimisation problem (5.2.2) where the control vector $\boldsymbol{u} = (c_l, c_t)$ contains two scalars representing a constant longitudinal and transversal wave speed.

Accurately knowing the wave speeds is essential for hybrid methods to be used for material characterisation and defect detection. E.g., when time reversal simulations are performed based on real data, uncertainties of 2% on the wave speeds are sufficient to cancel the refocused signal [86].

We start with a discussion of the stability of the problem followed by the introduction of a test case that will be used to compare several gradient free methods to a gradient based optimisation approach.

5.5.1 Stability of the problem

Before solving this inverse problem, we discuss the stability of the problem with respect to the controls, i.e., the different wave speeds in the problem. Inverse problems are often unstable, meaning that minor changes in the input, i.e., the observations, can have major influence on the output, i.e., the unknown wave speed. A rigorous stability analysis for the problem of wave speed determination is outside the scope of this work. We will however give an intuitive reasoning similar to reasoning presented by Kaipio and Somersalo [61] for a heat diffusion problem.

Consider a one dimensional semi-infinite domain $\Omega = \mathbb{R}^+$, with a transmitter at $x = 0$, a receiver at $x = X$ and constant wave speed c. As input of the inverse problem, we observe the particle velocity $v_i = v(t_i)$ at every time step t_i at the receiver. From this observation, we want to retrieve the wave speed. When a constant, spatially independent wave speed c is used, the only way to perturb the wave speed is to increase or decrease it. This results in a signal that arrives earlier or later, as the receiver picks up the signal of the source with a delay of X/c. The change in the observation varies smoothly with respect to the wave speed. Due to this sensitivity of the forward problem with respect to the wave speed, the inverse problem can be considered as stable.

5.5.2 Introduction of the test case

We introduce a test case for wave speed determination based on the observation of the normal particle velocity at one point on the boundary of an object consisting of a homogeneous, isotropic and linear material. We first define the underlying forward model used to obtain observations that will be the input of the inverse problem. Next, we formulate the inverse problem.

We consider the two dimensional elastodynamic example depicted in Figure 5.1 with computational domain $\Omega = [0, 40]\,\text{mm} \times [0, 20]\,\text{mm}$ and boundary conditions

$$\begin{aligned} \boldsymbol{v} &= \left(0, v_y\right) \quad \text{on} \quad \partial\Omega \cap \{y = 0\text{mm}\}, \\ \boldsymbol{\sigma} &= 0 \quad \text{on} \quad \partial\Omega \backslash \{y = 0\text{mm}\}, \end{aligned} \tag{5.22}$$

where

$$v_y = \exp\left(-\left(\frac{x - S_x}{3\text{mm}}\right)^2\right)\left[\frac{1}{2} + a(t - t_0)^2 \exp\left(a(t - t_0)^2\right)\right], \tag{5.23}$$

where $S_x = 20$mm is the position of the centre of the source, $t_0 = 2 \times 10^{-6}$ s is the pulse offset, the parameter $a = -\left(\pi f_c\right)^2$ and the central frequency $f_c = 600$ kHz. Figure 5.2 shows the excitation (5.23) evaluated at the centre of the source $x = S_x$. The wave speeds are chosen equal to the wave speeds in aluminium, i.e., $c_{l,\text{real}} = 6323$ m/s and $c_{t,\text{real}} = 3137$ m/s. Unless mentioned otherwise, we consider the time interval $t \in [0, 10]$ μs, which is plenty of time for the waves to travel from the source to the receiver R located at $(30, 20)$ mm. The observation is computed on a mesh with 21 × 11 crossed-triangle cells[6] using second order discontinuous elements. The time integration is done with the classical leapfrog method and a time step $\Delta t = 10^{-8}$s. Figure 5.3 shows a snapshot of the magnitude of the computed particle velocity $\boldsymbol{v}$ on the computational domain at time $t = 3\mu$s.

The input of the inverse problem is

$$v_{\text{obs}}\left(t_i\right) = v_y(R, t_i) + \eta_i, \tag{5.24}$$

where η_i is Gaussian noise $\sim \mathcal{N}\left(0, \gamma^2 \mathbf{I}\right)$, where for this problem the standard deviation $\gamma = 0.01 \times \max_t \left|v_{\text{obs}}\left(t\right)\right| \approx 0.005$m/s[7]. For the inverse problem a mesh with 13 × 7 crossed-triangle cells and third order elements are used. By doing so, we have avoided the inverse crimes mentioned in Section 5.2.4. The time step size for the inverse problem is the same as for the forward problem.

The inverse problem aims to retrieve the longitudinal and transversal wave speeds c_l and c_t respectively. The material is assumed to be homogeneous and isotropic, hence the mass density can be determined experimentally accurately up to measurement error and is assumed to be known. In practice, we have to keep some uncertainties in mind: When a piezoelectric transducer is used to excite the sample, the relation of the applied electrical signal and the actual particle displacement might only be known roughly. Furthermore, in most materials the waves are attenuated while propagating through the sample. We will not take these uncertainties into consideration for the execution of the simulations, but will discuss the effect they may have on the proposed procedures afterwards.

5.5.3 Time of flight techniques

A first family of techniques for wave speed determination are so-called *time of flight* methods, which exploit the fact that the time needed for a wave to travel from A to B equals the travelled distance divided by the average wave speed. As we are focussing on wave speed determination, the idea is to measure the time a wave needs to travel from a known location in a straight line to a known destination. A straight path between transmitter and receiver must exist, which is not always realistic. For this method, the source and the attenuation in the model do not have to be known. Time of flight methods do not involve the

---

[6] Crossed-triangle cells are formed by using the diagonals of a rectangular cell to obtain four triangular cells as used in Figure 5.3.

[7] One percent noise is realistic for measuring a signal in a 20V range using 11 bit accuracy. As using 16 bits is quite common, this is a significant amount of noise. Using 8 bits would result in 7% noise.

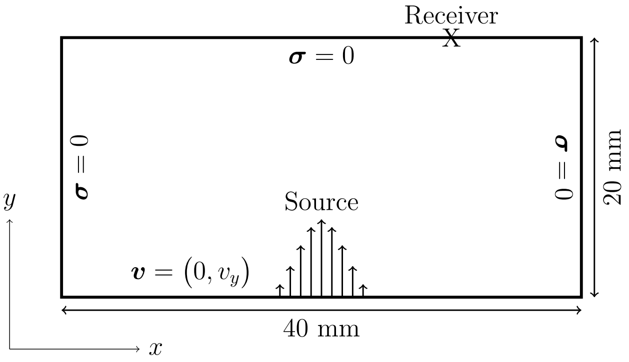


Figure 5.1: Set-up for the test case used in Section 5.5.

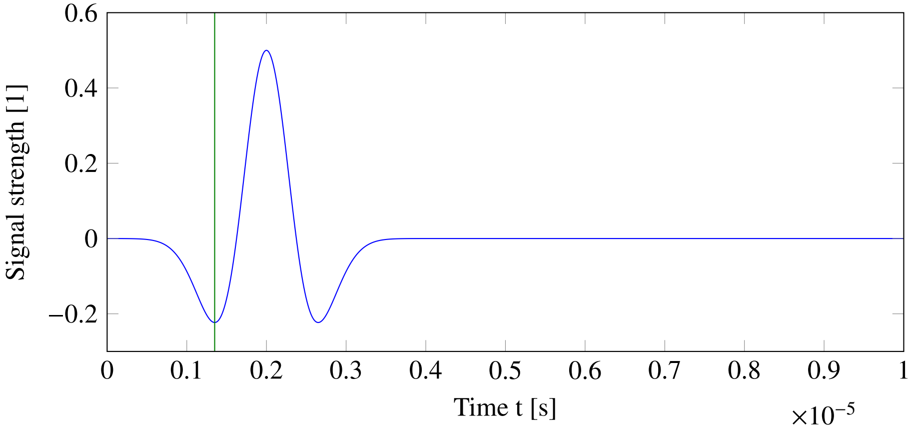


Figure 5.2: Ricker wavelet used as excitation in the forward problem of the test case used in Section 5.5. The green line marks the time instance at which the first minimum of the signal occurs, which is used in the time of flight method presented in Section 5.5.3.

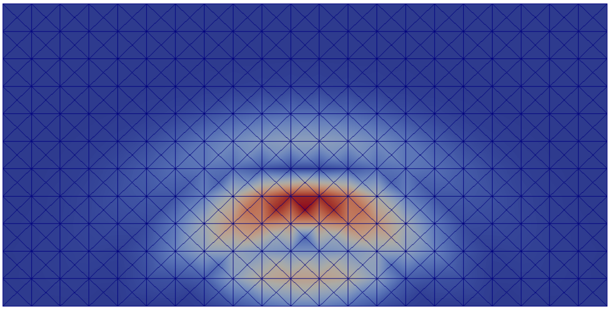


Figure 5.3: Snapshot of the computed particle velocity $\boldsymbol{v}$ at time $t = 30\mu s$ in the forward problem of the test case used in Section 5.5.

solution of a forward problem and can be considered a purely experimental technique.

Marking the exact moment of arrival of an emitted signal is difficult due to noise and diffraction. To solve this problem a pulse can be used as transmitted signal, which allows measuring the time between emission and arrival of the peak of the pulse. The peak will be somewhat shifted due to interference caused by reflections and the fact that there are two wave speeds. Note that when a peak is used to mark the arrival of a signal, we have to know when this peak was emitted.

Noise, measurement error and the chosen sample rate for the measurement are some of the secondary effects that make time of flight methods insufficiently accurate. The lack of accuracy motivates the use of hybrid methods.

Test case

We use the time of flight method to determine a first guess for the unknown wave speeds. For this, we look at the signal obtained by the receiver as shown in Figure 5.4. We focus on the arrival of the first minimum, emitted at $t = 1.35\mu s$, indicated by the vertical green line in the emitted signal shown in Figure 5.2. Figure 5.4 shows the expected arrival time, when using the correct longitudinal and transversal wave speed, of this minimum in the longitudinal and transversal

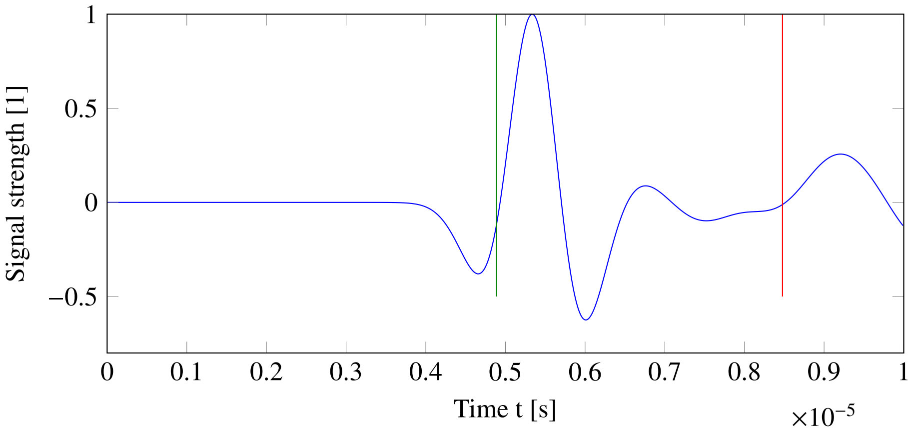


Figure 5.4: Normalised received signal in the forward problem of the test case used in Section 5.5.

wave by a vertical green and red line respectively. A first observation is that there is no observable peak for the transversal wave. Even for this fairly simple problem it appears to be difficult to determine the transversal wave speed with a time of flight method. The longitudinal wave is however clearly observed, be it somewhat disturbed due to reflections. We see that the first minimum appears to arrive early at $t = 4.66\mu s$.

Since the wave has travelled $\sqrt{2^2+1^2} = \sqrt{5}$cm in $4.66 - 1.35 = 3.31\mu s$, we estimate the longitudinal wave speed c_l to be 6756m/s. As a guess for the transversal wave speed we take half the longitudinal wave speed [65], i.e., $c_t = 0.5c_l = 3378$m/s. This means that the error on our first approximations is quite large: 7% on the longitudinal wave speed and 8% on the transversal wave speed. As Scalerandi et al. [86] have shown that a 2% error on the wave speeds is sufficient to cancel a time reversal focus, this approximation is unsatisfactory and has to be improved to be useful in further experiments.

5.5.4 Sampling the parameter space

As we are studying a problem with only two unknown parameters, sampling the parameter space is a second option to solve the problem. Assume we have a crude approximation of the longitudinal and transversal wave speed c_l and c_t, e.g., obtained with a time of flight procedure. Then we can restrict the parameter space to $[c_l - \delta_l, c_l + \delta_l] \times [c_t - \delta_t, c_t + \delta_t]$, where δ_l and δ_t are error margins. Next, we sample this function space with N_s samples in both intervals, leaving us with N_s^2 pairs of wave speeds to test. To get a sufficient accuracy for the retrieved values, N_s has to be high enough. Another option to increase accuracy is to use this technique iteratively, by restricting the parameter space around the most likely candidate of the previous iteration.

To evaluate how close a sampled pair of wave speeds is to the exact pair, we compute the misfit

$$J = \sum_{j=1}^{N_t} \frac{1}{2} \Big(\boldsymbol{v} \left(R, t_j \right) - \boldsymbol{v}_{\text{obs}} \left(R, t_j \right) \Big)^2 , \tag{5.25}$$

where $\boldsymbol{v}\left(R, t_j\right)$ and $\boldsymbol{v}_{\text{obs}}\left(R, t_j\right)$ respectively are the simulated and observed normal component of the particle velocity at the receiver position R and time instances t_j, $j = 1 \ldots N_t$.

This method does require the source and the attenuation to be known quite accurately. By normalising $\boldsymbol{v}\left(R, t_j\right)$ and $\boldsymbol{v}_{\text{obs}}\left(R, t_j\right)$, we can use this method without knowing the amplitude of the source and the attenuation. We could include both parameters as controls of the problem, but that would lead us to sample a control space of at least dimension four[8] instead of a two dimensional space. Assume $N_s = 10$, adding two additional controls would increase the number of times the forward problem has to be solved from 100 to 10^4.

Test case

Based on the approximations for the wave speeds obtained with the time of flight method in Section 5.5.3, we sample the parameter space and compute the misfit (5.25) for the chosen samples. We choose the samples in a structured manner, namely by using the guess $(c_l, c_t) = (6756, 3378)\,\text{m/s}$ with $0, 2, 4, 6, 8$ and 10 percent deviation, resulting in 11×11 samples in total. Figure 5.5 shows the misfit between the received signal and the computed result for the 11×11 samples. The minimal misfit is obtained for the $(c_l, c_t) = (6350, 3175)\,\text{m/s}$.

We execute a second iteration by using the updated guess $(c_l, c_t) = (6350, 3175)\,\text{m/s}$ and a finer resolution of $0, .5, 1, 1.5$ and 2 percent deviation, resulting in 9×9 samples in total. Figure 5.6 shows the misfit between the received signal and the computed result for the second batch of 9×9 samples.

The final approximation $(c_l, c_t) = (6318, 3143)\,\text{m/s}$ was obtained with a total of $(11 \times 11 + 9 \times 9) \times 1000 = 202,000$ time steps, and has a maximum error of 0.25% compared to the exact solution $(c_l, c_t) = (6323, 3137)\,\text{m/s}$.

---

[8] The attenuation is frequency dependent and differs for longitudinal and transversal waves, hence multiple parameters could be used to properly model attenuation. In practice, however, attenuation is often modelled by a constant.

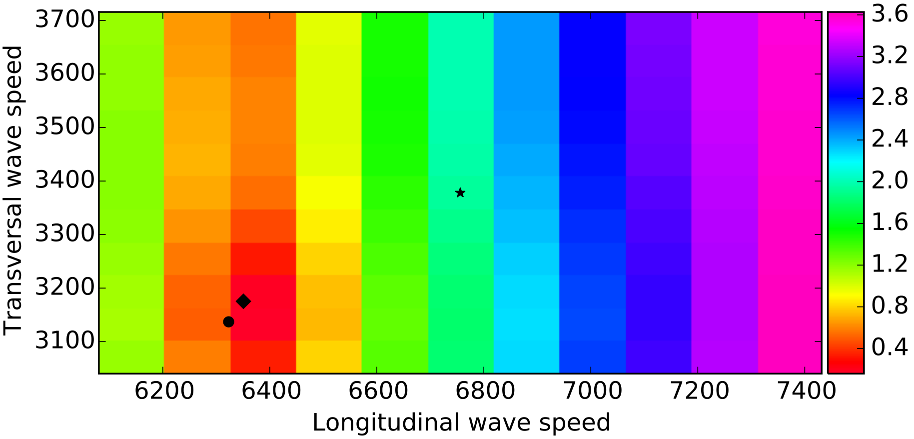


Figure 5.5: Misfit between the received signal in the forward problem of the test case used in Section 5.5 and the computed result with 11×11 samples by using the guess $(c_l, c_t) = (6756, 3378)$ m/s with $0, 2, 4, 6, 8$ and 10 percent deviation. The star marker indicates the initial guess, obtained with the time of flight method. The circle indicates the actual wave speeds, and the diamond indicates the updated guess.

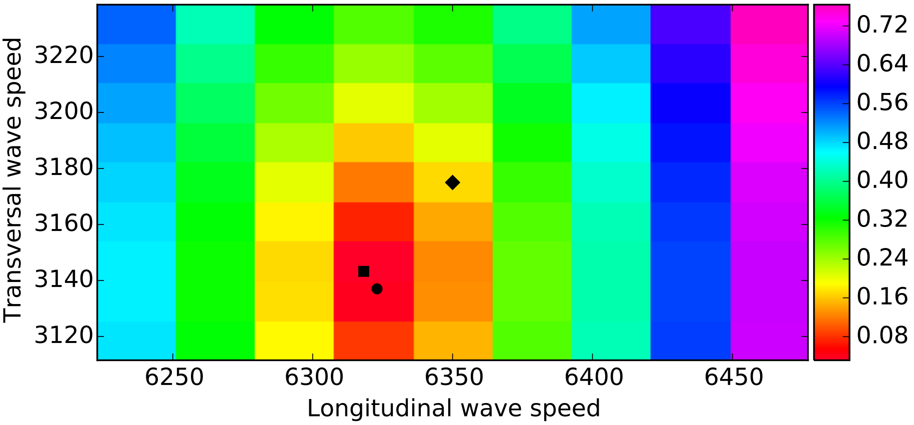


Figure 5.6: Misfit between the received signal in the forward problem of the test case used in Section 5.5 and the computed result with 9×9 samples by using the guess $(c_l, c_t) = (6350, 3175)$ m/s and a finer resolution of $0, .5, 1, 1.5$ and 2 percent deviation. The diamond marker indicates the initial guess, obtained from the previous iteration in Figure 5.5. The circle indicates the actual wave speeds, and the square marker indicates the updated guess.

5.5.5 Powell's method and Nelder-Mead

Next, we consider Powell's method and Nelder-Mead (see Section (5.3.2)) to find a minimiser for the misfit objective 5.25. Both methods do not use gradient information. In contrast to the preceding methods, there is no need for a sample distance nor a restriction on the admissible set[9]. For these automatic optimisation methods, the source and attenuation need to be known quite accurately. These methods suffer less from the curse of dimensionality than a sampling technique. Studying the behaviour of these methods for an increasing number of unknowns will however not be studied in this work.

Test case

Based on the approximations for the wave speeds obtained with the time of flight method in Section 5.5.3, we again use the initial guess $(c_l, c_t) = (6756, 3378)\,\mathrm{m/s}$. Powell's method results in the estimate $(6315, 3139)\,\mathrm{m/s}$ after sixty-two iterations which has an error below 0.2% compared to the exact solution $(6323, 3137)\,\mathrm{m/s}$. For this estimate $62 \times 1000 = 62,000$ time steps were solved.

Nelder-Mead results in the estimate $(6323, 3130)\,\mathrm{m/s}$ after thirty iterations which has an error below 0.1%. For this estimate only $30 \times 1000 = 30,000$ time steps were solved. As expected, both methods perform better than sampling the parameter space.

5.5.6 Wave speed determination using time reversal

A drawback of the method in Sections 5.5.4 and 5.5.5 is that the emitted signal has to be known. This can be avoided by using time reversal. A first idea is to sample the control space as before, but instead of using a misfit objective (5.9), we use reciprocal time reversal to evaluate the quality of the guesses for c_l and c_t. For this, we compute the response at the receiver when the observed signal is time reversed and emitted from the transmitter. The better the guess for the wave speeds in the computation, the clearer the refocused signal will be. We measure the quality of the refocused signal by the maximum amplitude occurring in the received signal.

To see why the maximum amplitude is a good measure, imagine we record the signal everywhere on the boundary of the domain, and during the time reversal

---

[9] By choosing a sample distance and a number of samples, the admissible set is defined by the selected samples.

phase, emit all measured signals reversed in time. If the wave speeds are chosen correctly, the source will be reconstructed. If the used wave speeds are incorrect, the signals from different parts of the boundary will arrive back at the source at different times, preventing constructive interference, leading to an overall lower maximal amplitude. If only one receiver is used for the time reversal phase with wrong wave speeds and without reflections at the object's boundaries, the signal sent from that receiver will arrive at the source at a different time. The signal will however have the same maximal amplitude. Hence, multiple receivers or reflections, which can be interpreted as virtual receivers/sources, are essential to allow constructive interference. As we consider a problem where only one receiver is used, we need reflections of the signal at the object's boundaries. To capture sufficient reflections, the signal has to be recorded for a longer time.

A more sophisticated wave speed determination technique based on time reversal has been proposed by Janssen and Van Den Abeele [59]. In this technique, instead of sending back the received signal $R(t)$, a signal F composed of a range of compressed and dilated signals that are superimposed on the received signal is sent back. The composed signal relies on two parameters as follows:

$$F\left(t, Z, \Delta\gamma\right) = R(t) + \sum_{k=1}^{Z} \left[R((1 + k\Delta\gamma)\, t) + R((1 - k\Delta\gamma)\, t)\right], \tag{5.26}$$

where Z determines the number of superimposed signals and $\Delta\gamma$ is the compression/dilation factor.

By using the composed signal (5.26) for the time reversal, the wave speeds can be retrieved by updating an initial guess using the following procedure:

Step 1: Find an approximation for the velocity ratio by searching for a maximal focus quality along a line of constant $c_l = c_{l,\text{guess}}$ or constant $c_t = c_{t,\text{guess}}$ using the composed signal $F\left(t, Z, \Delta\gamma\right)$ as input for the reciprocal time reversal simulation.

Step 1a: Perform N_s simulations along a line $c_l = c_{l,\text{guess}}$ in (c_l, c_t)-space, and determine the coordinate of the maximum value. Call this coordinate $c^*_{t,\text{guess}}$.

Step 1b: Perform N_s simulations along a line $c_t = c^*_{t,\text{guess}}$ in (c_l, c_t)-space, and determine the coordinate of the maximum value. Call this coordinate $c^*_{l,\text{guess}}$.

The corresponding velocities provide an approximation for the velocity ratio $c_{t,\text{real}}/c_{l,\text{real}} = c^*_{t,\text{guess}}/c^*_{l,\text{guess}}$, as argued by Janssen and Van Den Abeele [59].

Step 2: Find an approximation for the velocities: Approximations for $c_{t,\text{real}}$ and $c_{l,\text{real}}$ are achieved by searching for the maximum focus quality along the approximate diagonal $c_t = \left(c^*_{t,\text{guess}}/c^*_{l,\text{guess}}\right) c_l$ using the original signal $R(t)$ for the time reversal (implying that a focus can only occur at the real velocities). This second step requires another N_s simulations.

Three methods have been proposed to apply this procedure iteratively to improve the retrieved values:

1. Start from a large Z and low $\Delta\gamma$. Lowering the value for Z in every iteration and keeping $\Delta\gamma$ constant.
2. Reduce the spacings Δc_l and Δc_t for a constant Z, $\Delta\gamma$ and constant number of samples N_s resulting in a smaller search domain and higher resolution.
3. Start from a large Z and low $\Delta\gamma$. Lowering the product $Z\Delta\gamma$ by decreasing Z and increasing $\Delta\gamma$.

Based on the results from a first iteration, the search intervals for c_l and c_t can be reduced. Janssen and Van Den Abeele [59] concluded that all iteration methods perform equally well.

In summary, we point out that the main benefit of using time reversal is that the source signal nor the attenuation in the model are required for obtaining an approximation of the wave speeds. Testing a pair of samples (c_l, c_t) takes however more time, as a longer time interval is required than in the previous methods. By exploiting the properties of time reversal, the number of samples can be reduced in comparison with uniform sampling from N_s^2 to $3N_s$.

Test case

Based on the approximations for the wave speeds obtained with the time of flight method in Section 5.5.3, we sample the (c_l, c_t) parameter space and determine the maximum amplitude of the time reversed signals for the chosen samples. We again use the guess $(c_l, c_t) = (6756, 3378)\,\text{m/s}$ with $0, 2, 4, 6, 8$ and 10 percent deviation, resulting in 11×11 samples in total. Figure 5.7 shows the maximum amplitude in the time reversed signals when a time interval of $10\mu\text{s}$ is used as before. The maximum amplitude is obtained for the $(c_l, c_t) = (7161, 3242)\,\text{m/s}$, which is a significantly worse approximation than the initial guess. This was to be expected, as the used time interval is too small to allow reflections.

Figure 5.8 shows the maximum amplitude in the time reversed signals when a twice as long time interval of 20μs is used. The maximum amplitude is obtained for the $(c_l, c_t) = (6350, 3175)\,\mathrm{m/s}$, which is the same estimate as we obtained when using the misfit objective in Section 5.5.4. The difference is that we did not use the forward signal in this procedure at the cost of having to solve twice as many time steps due to the longer time interval. More specifically, $11 \times 11 \times 2000 = 242,000$ time steps were computed for this approximation.

Next, we use the more sophisticated approach proposed by Janssen and Van Den Abeele [59] for a time interval of 20μs. Starting from the guess $(c_l, c_t) = (6756, 3378)\,\mathrm{m/s}$, a sample distance of 2%, we obtain the updated estimate $(c_l, c_t) = (6356, 2979)\,\mathrm{m/s}$ and composed signal $F(t, Z, \Delta\gamma) = F\left(t, 64, 0.0625\%\right)$[10]. The approximation for c_l is quite good, but the approximation for c_t has hardly improved. A second iteration, where we reduce the sample distance leads to the estimate $(c_l, c_t) = (6547, 3009)\,\mathrm{m/s}$, which is worse than the previous estimate. As these results are not satisfying, we repeat the experiment for a time interval of 80μs. Using the same settings as before, we get the estimate $(c_l, c_t) = (6080, 3175)\,\mathrm{m/s}$ after one iteration and $(c_l, c_t) = (6323, 3301)\,\mathrm{m/s}$ after two iterations. These results are still not satisfying, even though $3 \times 11 \times 8000 = 264,000$ time steps were needed per iteration, which is considerably more than both previous methods.

There is room for further investigation with this method, e.g., by trying other parameters for the composed signal. As we are aiming for *automated* procedures that do not require trial and error approaches to be fine tuned for specific problems, we decide to abandon this method for further investigation.

5.5.7 Wave speed determination using gradient information

The last method we consider for wave speed determination is the BFGS method (see Section 5.4.1), which uses gradient information. Just like the methods in Powell's method and Nelder-Mead, this method only needs an initial guess to start. Also for this automatic optimisation method, the source and attenuation need to be known quite accurately. In comparison to the sampling method discussed in Section 5.5.4, adding a control for the attenuation and the amplitude of the source will result in little additional computations per iteration.

For this problem, we again look for a minimiser of the misfit objective (5.25)

$$J = \sum_{j=1}^{N_t} \frac{1}{2} \left(\boldsymbol{v}_{\text{computed}} \left(R, t_j\right) - \boldsymbol{v}_{\text{obs}} \left(R, t_j\right) \right)^2 . \tag{5.27}$$

---

[10] We use the most frequently used values for Z and $\Delta\gamma$ in [59].

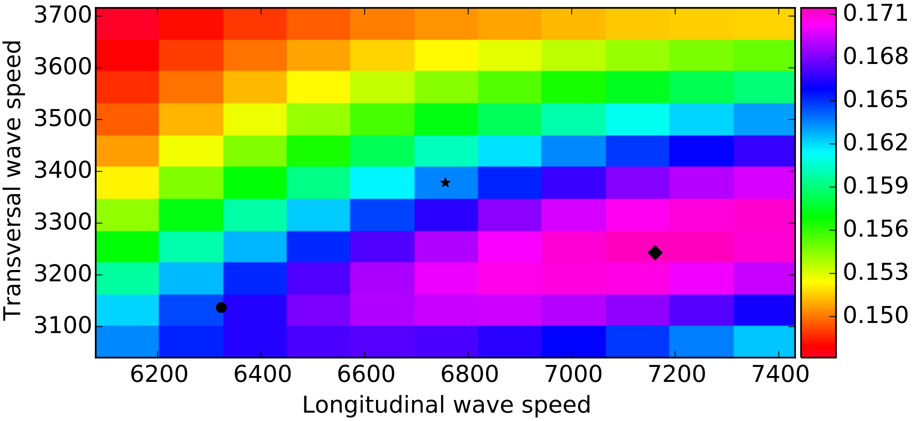


Figure 5.7: Maximum amplitude of the computed time reversed signal with 11 × 11 samples, emitting the received signal from the forward problem of the test case used in Section 5.5. Using a time interval that is too short for reflections to occur. The star marker indicates the initial guess, obtained with the time of flight method. The circle indicates the actual wave speeds, and the diamond indicates the updated guess.

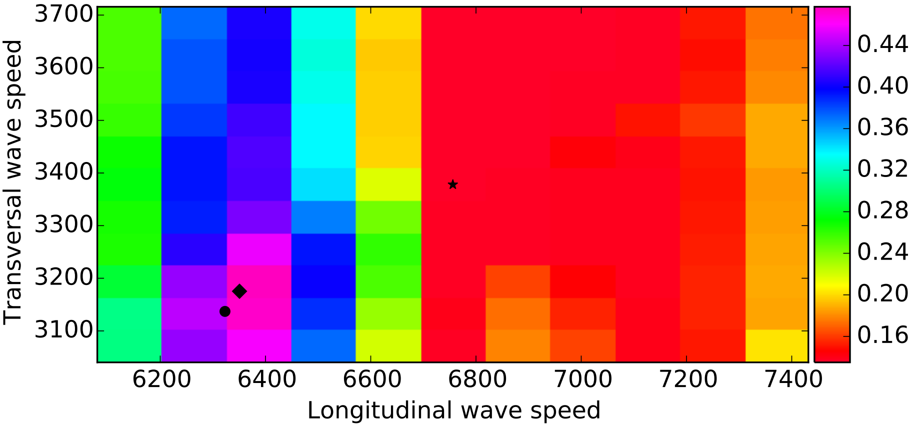


Figure 5.8: Maximum amplitude of the computed time reversed signal with 11 × 11 samples, emitting the received signal from the forward problem of the test case used in Section 5.5. Using a time interval that is large enough for reflections to occur, more specifically the time interval for these results is twice the size of the interval used for the results in Figure 5.7. The star marker indicates the initial guess, obtained with the time of flight method. The circle indicates the actual wave speeds, and the diamond indicates the updated guess.

As motivated in Section 5.5.1, we assume this problem is sufficiently stable. Hence, we do not use regularisation for this problem.

Derivative of the objective

The gradient will be computed automatically from the forward problem by the dolfin-adjoint library [38, 44]. However, we need to implement the objective functional J and its derivative $\frac{\partial J}{\partial \boldsymbol{q}}$ manually.

The derivative of the objective (5.25) with respect to the states $\boldsymbol{q}$ is

$$\frac{\partial J}{\partial \boldsymbol{q}} = \Psi\left(R\right) \sum_{j=1}^{N_t} \left(\boldsymbol{q}_{\text{computed}}\left(R, t_j\right) - \boldsymbol{q}_{\text{obs}}\left(R, t_j\right) \right), \tag{5.28}$$

where

$$\Psi\left(R\right) = \left[\phi_0\left(R\right), \phi_1\left(R\right), \ldots, \phi_{N_d-1}\left(R\right), \phi_{N_d}\left(R\right)\right], \tag{5.29}$$

is a vector that contains all trial functions ϕ_i evaluated in the receiver R.

Evaluating trial functions at specific coordinates should be done with caution. When evaluating discontinuous basis functions in spatial discretisation points on interior facets, multiple values may conflict. To avoid any ambiguity, we make sure that receiver coordinates do not coincide with shared discretisation points.

5.5.8 FEniCS and dolfin-adjoint implementation

We start from a FEniCS implementation for the forward problem

```
v, times = forward(cl, ct, Annotate = True)
```

that returns the computed velocity field `v` at final time and the list of time instances `times` at which intermediate results were computed, for a specific longitudinal wave speed `cl` and transversal wave speed `ct`. The implementation of this `forward` routine was explained in Chapter 4 and will not be repeated here. A routine to determine the unknown wave speeds looks like the subsequent code.

```
from dolfin import *
from dolfin-adjoint import *
...
```

```
# Set starting values for the unknown parameters
cl = Constant(6756., name = "cl")
ct = Constant(3378., name = "ct")

# Execute the forward solver to annotate and record the tape
v, times = forward(cl, ct, Annotate = True)

# Define the control variables
ccl = Control(cl)
cct = Control(ct)

# Prepare the objective function
J = PointwiseFunctional(v, [refs[1:]], R, times[1:], v_ind = [1])

# Prepare the reduced functional
reduced_functional = ReducedFunctional(J, [ccl, cct])

# Run the optimisation
minimize(reduced_functional, method = "L-BFGS-B",\
tol=1.0e-18, options = {"disp": True,"gtol":1.0e-18})
```

In this program, the forward solver is annotated by dolfin-adjoint to record the so called *tape*. This tape is used to compute the forward and adjoint states of the inverse problem for the consecutive iterations of an optimisation routine.

The `PointwiseFunctional` object was implemented as part of this work and has been added to the dolfin-adjoint library. This object returns an objective functional where the `v_ind` component of a computed vector field `v` evaluated at the spatial coordinates `R` is compared to an observed reference value `refs` for every time instance in the list `times`. In other words, this object is the implementation of the objective (5.27). This functional can then be used to construct the `ReducedFunctional`, which is then passed to a minimiser, in this case the `L-BFGS-B` minimiser.

For complete and working dolfin-adjoint code relevant to this work, we refer to the code of Chapter 6, which is available on-line as supporting material [96].

Test case

Based on the approximations for the wave speeds obtained with the time of flight method in Section 5.5.3, we again use the initial guess $(c_l, c_t) = (6756, 3378)$ m/s. Figure 5.9 shows the consecutive estimates of the BFGS process on top of

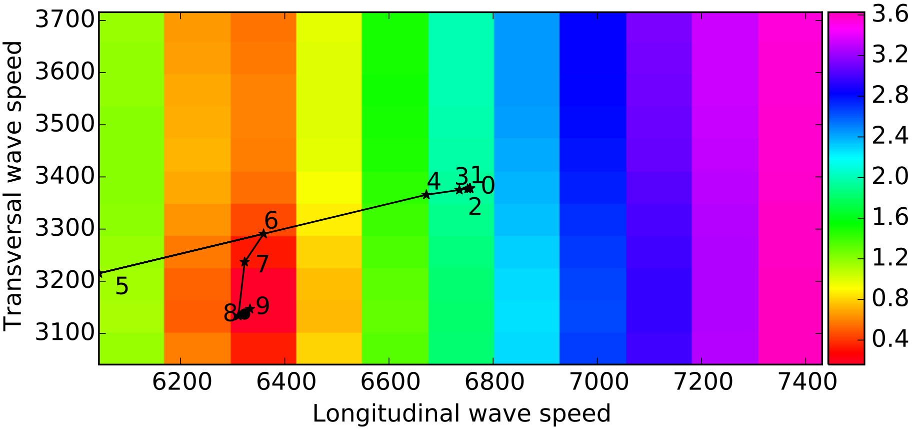


Figure 5.9: Misfit between the received signal in the forward problem of the test case used in Section 5.5 and the computed result with 11 × 11 samples by using the guess $(c_l, c_t) = (6756, 3378)$ m/s with 0, 2, 4, 6, 8 and 10 percent deviation as presented in Section 5.5.4. The star markers indicate the consecutive samples computed in the BFGS optimisation process. The circle indicates the actual wave speeds.

the results from Section 5.5.4. After ten iterations, we obtain the estimate $(c_l, c_t) = (6333, 3146)$ m/s which has an error below 0.2%. For this estimate only $2 \times 10 \times 1000 = 20,000$ time steps were solved, which is an order of magnitude less than the techniques in Section 5.5.4 and 5.5.6. Powell's method and Nelder-Mead needed respectively 200% and 50% more time steps to achieve a similar accuracy.

5.5.9 Discussion and conclusion of test case results

In the previous sections we have discussed several methods to determine the transversal and longitudinal wave speeds in homogeneous isotropic media. We have applied the methods to the test case presented in Section 5.5.2.

In the subsequent section we will study the more complicated inverse scattering problem where the unknown parameters are spatially dependent material parameters. We therefore evaluate the results of the different methods applied to the test case with homogeneous and isotropic material in this section and determine the applicability and potential of the different methods to the inverse scattering problem.

Time of flight methods only allow to estimate the average wave speeds. In practice, it proved difficult to estimate more than only the largest wave speed, and even this resulted in an estimate with poor accuracy. Uniformly sampling the parameter space suffers from the curse of dimensionality, which makes sampling infeasible to be used for a larger number of controls. The more sophisticated time reversal approach from [59] is only applicable in the specific case of homogeneous and isotropic material. It proved to be the least effective of all considered methods for this problem. Many alternative, problem specific, methods that rely on time reversal exist, among which also methods to locate inhomogeneities.

The gradient based approach is not problem specific and was the most effective for the wave speed determination case. The gradient free Nelder-Mead method is also not problem specific and was almost as fast as the gradient based approach. However, as noted in Section 5.3.2, Nelder-Mead will have a very large set-up cost for spatially dependent problems. Hence, only the gradient based approach will be considered as an option to solve the inverse scattering problem.

5.6 Inverse scattering problem

In this section, we study how to solve inverse scattering problems where spatially dependent material properties need to be determined by indirect observations. As argued in Section 5.5.9, gradient based approaches are best suited to handle this problem. The main difference with the problem and the method in Section 5.5.7 is that the controls are now functions instead of a vector of scalars.

5.6.1 Stability of the problem

To discuss the stability of the inverse scattering problem, we revisit the one dimensional set-up used for the stability discussion of the wave speed determination problem used in Section 5.5.1. Instead of a constant wave speed, we now consider a spatially varying wave speed $c(x)$. In this case, many different perturbations are possible, namely changing the wave speed locally at one or several positions.

Many different wave speed functions lead to the same average wave speed

$$\langle c \rangle = \frac{1}{X} \int_0^X c(x)\,\mathrm{d}x, \tag{5.30}$$

where $[0, X]$ is the considered spatial interval. As the time of flight method is only capable of determining the average wave speed, it is a very unstable method to determine spatially dependent wave speeds. Figure 5.10 shows several spatially varying relative wave speed distributions[11] $c_i(x)$ that have the same average. All these functions would be equally likely for a certain average wave speed determined by a time of flight method. Hence, more information has to be taken into account by the inverse solution method to determine a spatially dependent control.

Perturbing the wave speed functions leads to inhomogeneities which cause reflections. As reflections travel an indirect and hence longer path than the direct signal, they will only appear in the recordings after a certain amount of time T_r. When the objective function takes all recorded information into account up to time $t = T_r$, we will already be capable of distinguishing between the different wave speeds functions in Figure 5.10 since the amplitudes of the received signal will differ depending on the shape and number of perturbations in the wave speed function.

For any of the $c_i(x)$ depicted in Figure 5.10, a different position of the perturbation in the considered spatial interval will result in identical recordings for $t < T_r$. This still makes it hard for the inverse solver to determine which spatially varying wave speed function is the correct one. When the objective also considers observations for time $t > T_r$, then the arrival time of the reflections will be different, stabilising the inverse problem somewhat. The amplitude of the reflections, will however be small compared to the amplitude of the transmitted signal. Hence, a slight error in the observation drastically changes the likelihood of the different candidates, making this an unstable inverse problem.

5.6.2 Introduction of the test case

We consider the two dimensional square domain depicted in Figure 5.11 with computational domain $\Omega = [0, 20]\,\text{mm} \times [0, 20]\,\text{mm}$ and boundary conditions

$$\begin{aligned} \boldsymbol{v} &= (0, v_1) \quad \text{on} \quad \partial\Omega \cap \{y = 0\text{mm}\}\,, \\ \boldsymbol{v} &= (v_2, 0) \quad \text{on} \quad \partial\Omega \cap \{x = 0\text{mm}\}\,, \\ \boldsymbol{v} &= (0, v_2) \quad \text{on} \quad \partial\Omega \cap \{y = 20\text{mm}\}\,, \\ \boldsymbol{v} &= (v_1, 0) \quad \text{on} \quad \partial\Omega \cap \{x = 20\text{mm}\}\,, \end{aligned} \tag{5.31}$$

---

[11] We define the relative wave speed as the local wave speed divided by the minimal wave speed in the domain.

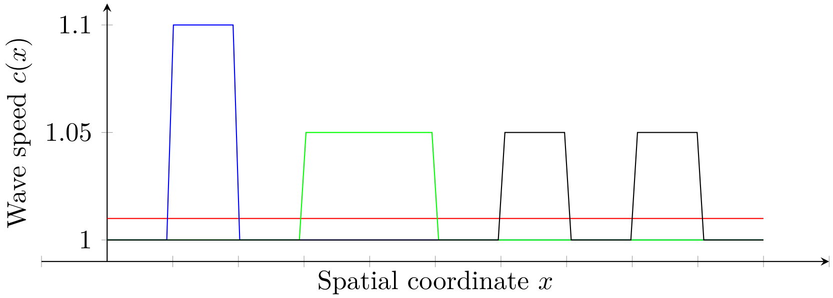


Figure 5.10: Three different spatially dependent wave speeds $c_i\left(x\right)$ as a function of the spatial position x indicated in blue, green and black. All wave speeds $c_i\left(x\right)$ have the same average indicated in red.

where

$$v_i = \exp\left(-\left(\frac{x - S_x^i}{2\text{mm}}\right)^2\right)\left[\frac{1}{2} + a(t-t_0)^2\exp\left(a(t-t_0)^2\right)\right], \tag{5.32}$$

where $S_x^1 = 5\text{mm}$, $S_x^2 = 15\text{mm}$ are the positions of the centre of the sources, $t_0 = 2 \times 10^{-6}$ s is the pulse offset, the parameter $a = -\left(\pi f_c\right)^2$ and the central frequency $f_c = 500$ kHz.

The transversal wave speed is chosen to be constant, i.e., $c_{t,\text{real}} = 3137$ m/s, and the longitudinal wave speed to be spatially varying. We will study a continuously and discontinuously spatially varying wave speed function. We consider the time interval $t \in [0, 10]$ µs, which is plenty of time for the waves to travel from the sources through heterogeneities. The observation is computed on a mesh with 11×11 crossed-triangle cells using second order discontinuous elements. The time integration is done with the classical leapfrog method and a time step $\Delta t = 10^{-8}$s.

The inverse problem aims to retrieve the spatially varying longitudinal wave speed function, assuming the transversal wave speed to be known. From the discussion in Section 5.6.1, we know that solving inverse scattering problems with spatially dependent wave speeds results in a loss of stability of the problem. The problem can be stabilised by using more receivers and transmitters or a longer time interval to increase the amount of information that is available to solve the inverse problem. To stabilise the problem and keep the simulation time low, we choose to record the solution everywhere in the computational

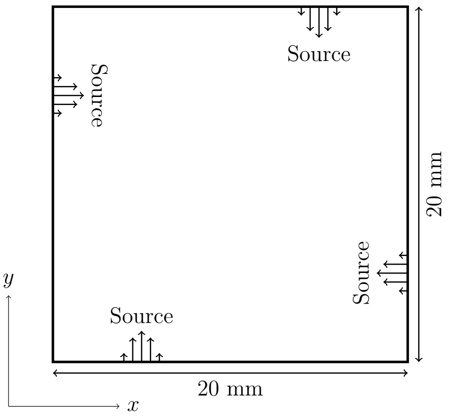


Figure 5.11: Set-up for the test case used in Section 5.6.2.

domain[12]. Hence we propose the following objective for the test case in this section

$$J = \sum_{j=1}^{N_t} \frac{1}{2} \left\| \boldsymbol{v}_{\text{num}}\left(t_j\right) - \boldsymbol{v}_{\text{obs}}\left(t_j\right) \right\|_{\Omega}^{2}. \tag{5.33}$$

A regularisation term is unnecessary, as the objective takes sufficient information into account to stabilise the problem. The inverse problem is solved using a mesh with 7 × 7 crossed-triangle cells using third order discontinuous elements. The longitudinal wave speed distributions are in any case discretised using third order discontinuous Lagrange elements.

Before we can study the results of solving inverse scattering problems with different unknown controls, we discuss the difficulties in solving inverse problems with functions as control variables in the subsequent sections.

5.6.3 Iterative methods in a Hilbert setting

From a practical point of view, a spatially dependent control is a collection of scalar controls, i.e., every degree of freedom is a scalar control. It is however important to also keep the theoretical point of view in mind, i.e., the control is a mathematical function with certain chosen properties as an approximation to a physical property. When applying iterative methods to variables that represent functions, proper preconditioners are required to obtain mesh independent

---

[12] We are aware that this is physically not possible, but in this work we are mainly interested in setting up a proper numerical experiment.

convergence rates [64, 69]. This is not only relevant when solving a linear system of equations, but also when solving PDE constrained optimisation problem where the controls are elements of a Hilbert space [37, 45]. Before we can accurately describe the problem, we need to introduce a *Hilbert space* and its dual[13].

Hilbert spaces

A Hilbert space is a complete inner product space. The dual of a Hilbert space X, denoted as X^*, is the space of all bounded linear functionals on X. By way of example, we consider two function spaces for the control functions, i.e., the Lebesgue space $L^2(\Omega)$ and the set $H^1(\Omega)$ which is the set of all functions in $L^2(\Omega)$ with weak derivatives in $L^2(\Omega)$.

The inner product for $L^2(\Omega)$ is

$$(u, v)_{L^2(\Omega)} = \int_\Omega uv \,\mathrm{d}x. \tag{5.34}$$

The inner product for $H^1(\Omega)$ is

$$(u, v)_{H^1(\Omega)} = \int_\Omega uv + \nabla u \cdot \nabla v \,\mathrm{d}x. \tag{5.35}$$

The norm $\|u\|_S$ of a vector u in either space S is defined through the inner product as

$$\|u\|_S = \sqrt{(u, u)_S}. \tag{5.36}$$

Optimisation with Hilbert controls

Objective functions like (5.33) are differentiable non-linear maps from a Hilbert space X to the set of real numbers, i.e., $J : X \to \mathbb{R}$. The Fréchet derivative of the objective with respect to the control is defined as

$$J'(u; v) = \lim_{\epsilon \to 0} \frac{J(u + \epsilon v) - J(u)}{\epsilon}. \tag{5.37}$$

The derivative of the objective evaluated at a fixed u (5.37) is a bounded linear map from X to $\mathbb{R}$, and hence an element of the dual Hilbert space X^*.

---

[13] We restrict ourselves to introducing the most essential concepts from functional analysis to explain the relevant phenomena. For a more rigorous introduction of the mentioned concepts, we refer to an introductory book on functional analysis such as, e.g., Rudin [83].

The descent step from the BFGS procedure explained in Section 5.4.1 looks like

$$x_{k+1} = x_k + \alpha_k J'\left(x_k\right). \tag{5.38}$$

In this step a primal value $x_k \in X$ and a dual value $J'\left(x_k\right) \in X^*$ are added, while this addition is not well defined. To resolve this issue, a pull-back operator, i.e., an operator that maps the dual space X^* to the primal space X, must be applied to the second term in (5.38). The canonical choice for this operator is the *Riesz map*. The Riesz representation theorem (see, e.g., [17]) states that any $j \in X^*$ can be uniquely represented by a $g \in X$ such that $j\left(u\right) = \left(g, u\right)_X$ for all $u \in X$.

In Section 5.4.1, we mention that Newton (type) optimisation methods use the Hessian of the objective function. When using a control in a Hilbert space, the second derivative of the objective is the Fréchet derivative of the first derivative (5.37) in a different perturbation direction p and is denoted by $J''\left(u; v, p\right)$. The second derivative $J''\left(u; v, p\right)$ is hence a map of the form $J'' : X \times X \times X \to \mathbb{R}$. The associated Newton-update (5.13) is determined by finding a specific descent direction $p^* \in X$ such that $J''\left(u; v, p^*\right) = -J'\left(u; v\right)$ for all $v \in X$. Hence we can see that the Hessian $\mathcal{H} = J''\left(u\right)$ is a map of the form $\mathcal{H} : X \to X^*$, such that

$$\langle \mathcal{H}(u) w, v \rangle = J''\left(u, v, w\right). \tag{5.39}$$

In this work we use the BFGS method as optimisation procedure which uses an approximation for the Hessian. In Equation (5.14), the Hessian approximation B_k in the BFGS procedure is applied to elements in the original Hilbert space X, resulting in an element of the dual Hilbert space X^* while the right hand side of (5.14) is a Riesz representer in X. In order for this equation to be well defined, B_k has to be preconditioned by the Riesz map associated with the used Hilbert space. This also applies to the initial guess, which hence should be the Riesz map instead of the identity matrix. Finally, we note that also the used line search procedure should be carefully implemented as it uses norms and angles that depend on the used Hilbert space. As we did not study the line search, we will not go into the details of this. We however use the Moola optimisation package [43], which properly handles these concerns.

Computing Riesz maps

To determine the Riesz representers needed to make the addition in (5.38) well defined, we use the Riesz representation theorem that states that any bounded linear functional $j \in X^*$ can be uniquely represented by a $g \in X$, via

$$\langle j, w \rangle = \left(g, w\right)_X, \qquad \forall w \in X, \tag{5.40}$$

where $\langle j, w\rangle = j(w) \in \mathbb{R}$ is the action of j on w. Specifically for the considered problem $j = J'(u; v)$.

For $X = L^2(\Omega)$, (5.40) expands to

$$J'(u; w) = \int_\Omega g\, w \,\mathrm{d}x = \langle Ag, w\rangle, \qquad \forall w \in X, \tag{5.41}$$

where A is the canonical operator associated with the bilinear form $(\,\cdot\,,\,\cdot\,)_X$. In (5.41) we recognise the Galerkin formulation of the problem

$$J'(u) = Ag, \qquad \text{in} \quad \Omega \tag{5.42}$$

Hence the Riesz representers g for control functions in $L^2(\omega)$ can be computed by solving the variational problem (5.41). The Riesz map $\mathcal{R} : X^* \to X$ is hence the inverse of the canonical operator A.

Similarly for $X = H^1(\Omega)$, (5.40) expands to

$$J'(u; w) = \int_\Omega g\, w + \nabla g \cdot \nabla w \,\mathrm{d}x, \qquad \forall w \in X. \tag{5.43}$$

In (5.43) we recognise the Galerkin formulation of the problem

$$J'(u) = g - \Delta g, \qquad \text{in} \quad \Omega \tag{5.44}$$

$$\nabla g \cdot \boldsymbol{n} = 0, \qquad \text{on} \quad \partial\Omega. \tag{5.45}$$

Hence the Riesz representers g for control functions in $H^1(\omega)$ can be computed by solving the variational problem (5.43).

5.6.4 Appropriate function space for the control function

In this section, we study the influence of the chosen function space for the control[14]. We compare the use of $L^2(\Omega)$ and $H^1(\Omega)$ as function space for the control with the use of the classical use of l_2 as space for the control. The assumption that the control is an element of l_2 implies that the underlying function space is ignored and that the control is treated as a vector of scalar controls.

We consider each function space for the control once for the smoothly varying longitudinal wave speed distribution shown in Figure 5.12a and once for

---

[14] An appropriate function space for the control should be chosen together with problem experts so that the mathematical function that represent the control can capture the desired physical properties.

the discontinuously varying longitudinal wave speed distribution shown in Figure 5.13a.

Figures 5.12 and 5.13 show the reconstructed value of the control for the smoothly and discontinuously varying wave speed distributions respectively using the three different Riesz maps. The left column shows the results after 10 BFGS iterations, while the right column shows the result after 70 or 100 BFGS iterations.[15] For visualisation, the third order discontinuous element approximation of the reconstructed controls have been projected onto first order continuous Lagrange element spaces.

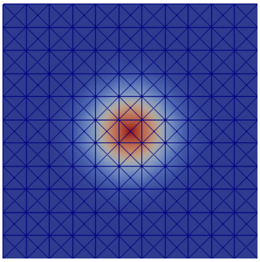
In both figures, we see that after 10 BFGS iterations with the $L^2(\Omega)$ and $H^1(\Omega)$ Riesz maps, the reconstructed wave speed distributions is closer to the reference solution than when using the l_2 Riesz map.

The reconstruction after 70 iterations for the smoothly varying defect when using the $H^1(\Omega)$ Riesz map is more accurate than when using the other Riesz maps. This was to be expected, as elements in $H^1(\Omega)$ are smoother than the elements of $L^2(\Omega)$. Hence, we expect for the discontinuously varying wave speed distribution that the $L^2(\Omega)$ Riesz map will perform best. This is however not clear from Figure 5.13 due to the fact that the chosen color map ranges from the minimum of the reference value (6320 m/s) to the maximum of the reference value (9480 m/s) while the maximum value of the different reconstructed controls exceeds these values. Hence we include a more accurate visualisation of the reconstructed control values in Figure 5.14 on a refined mesh. In this figure, each shown result has a different color map, for which the legend is included in the figure. From this refined visualisation of the reconstructed control, we can conclude that the solution using the l_2 Riesz map is again less accurate than when using $L^2(\Omega)$ and $H^1(\Omega)$ Riesz maps.

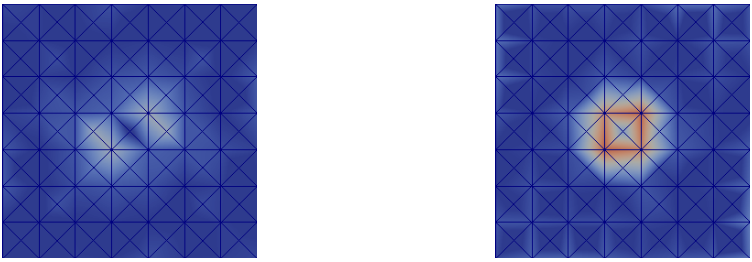
These results indicate that it is important to use proper function spaces for specific controls for two reasons. Firstly because a proper Riesz map will reduce the amount of required iterations and hence speed up the computation procedure. And secondly because the reconstruction process can converge to a more accurate solution.

---

[15] For the results with the $H^1(\Omega)$ Riesz maps, the right column shows the value of the control when the stopping criterion is reached, i.e., the difference $\left|J_k - J_{k+1}\right| \leq 10^{-10}$, where J_k is the value of the objective after k BFGS iterations. The BFGS procedure using the l_2 and $L^2(\Omega)$ Riesz maps was aborted after 100 iterations.

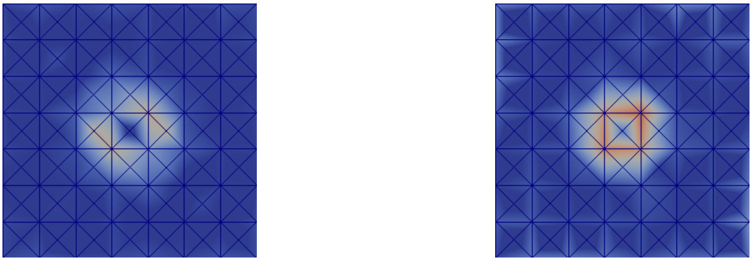


(a) Reference smooth wave speed distribution.



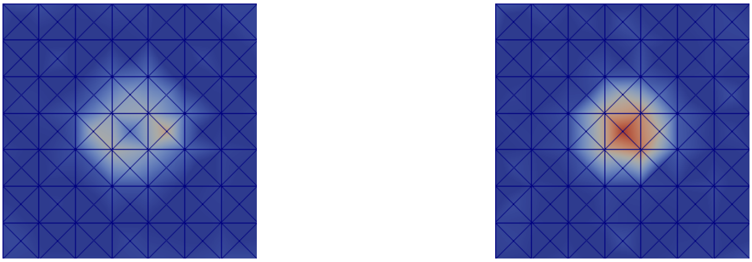
(b) Wave speed distribution recovered after 10 iterations with l_2 Riesz map.

(c) Wave speed distribution recovered after 100 iterations with l_2 Riesz map.



(d) Wave speed distribution recovered after 10 iterations with $L^2(\Omega)$ Riesz map.

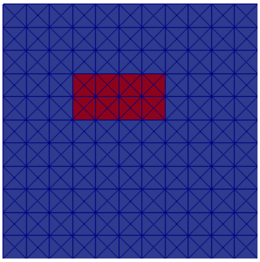
(e) Wave speed distribution recovered after 100 iterations with $L^2(\Omega)$ Riesz map.



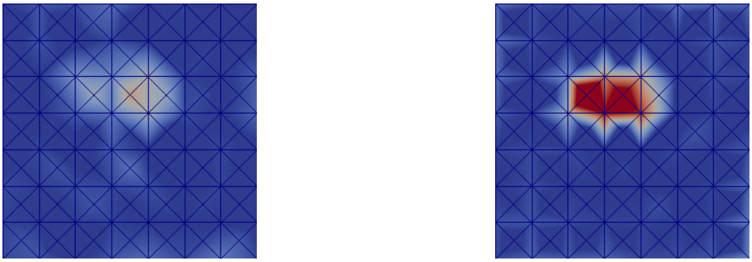
(f) Wave speed distribution recovered after 10 iterations with $H^1(\Omega)$ Riesz map.

(g) Wave speed distribution recovered after 70 iterations with $H^1(\Omega)$ Riesz map.

Figure 5.12: Wave speed distributions recovered by solving an inverse scattering problem ((b) to (g)) using the BFGS method and different Riesz maps. The smooth wave speed function (a) is the reference solution. The colour scale ranges from 6320 m/s (blue) to 9480 m/s (red).



(a) Reference discontinuous wave speed distribution.

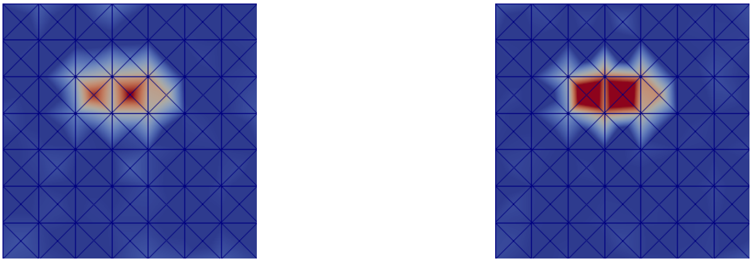


(b) Wave speed distribution recovered after 10 iterations with l_2 Riesz map.

(c) Wave speed distribution recovered after 100 iterations with l_2 Riesz map.

(d) Wave speed distribution recovered after 10 iterations with $L^2(\Omega)$ Riesz map.

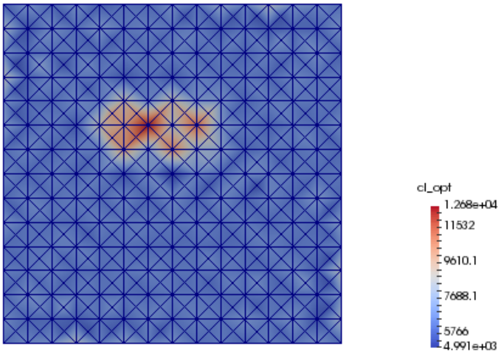
(e) Wave speed distribution recovered after 100 iterations with $L^2(\Omega)$ Riesz map.



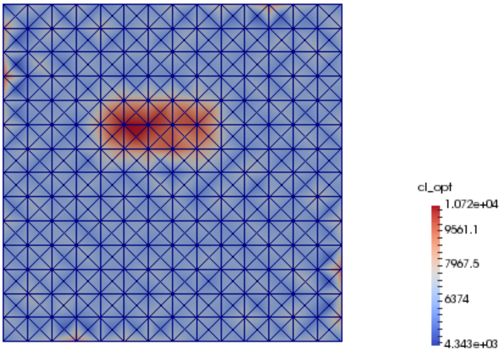
(f) Wave speed distribution recovered after 10 iterations with $H^1(\Omega)$ Riesz map.

(g) Wave speed distribution recovered after 70 iterations with $H^1(\Omega)$ Riesz map.

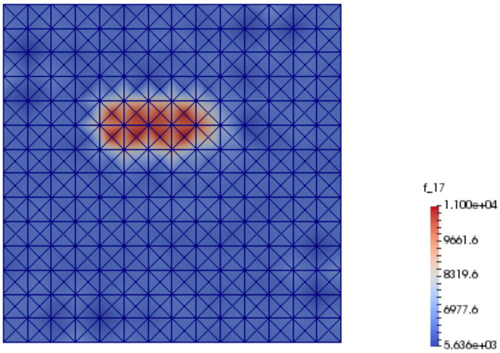
Figure 5.13: Wave speed distributions recovered by solving an inverse scattering problem ((b) to (g)) using the BFGS method and different Riesz maps. The discontinuous wave speed function (a) is the reference solution. The colour scale ranges from 6320 m/s (blue) to 9480 m/s (red).



(a) Wave speed distribution recovered after 100 iterations with l_2 Riesz map.



(b) Wave speed distribution recovered after 100 iterations with $L^2(\Omega)$ Riesz map.



(c) Wave speed distribution recovered after 70 iterations with $H^1(\Omega)$ Riesz map.

Figure 5.14: Reconstructed wave speed distributions as shown in Figures 5.13c, 5.13e and 5.13g on a refined mesh. The colour scale is mentioned for every result separately.

5.6.5 Outlook

In the previous sections, we indicated how to solve problems with functional controls and the importance of choosing suitable Hilbert spaces for the controls. With this information, it is possible to study inverse scattering problems in depth; further study of these problems will however not be included in this work. In this section, we give two suggestions for further study.

The first suggestion is how to proceed to a more realistic objective function, as assuming the solution is recorded everywhere in the domain is physically not possible. Ultimately the objective should contain discrete contributions of receivers on the boundary of the domain, as done for the wave speed determination problem in Section 5.5. This will however cause the problem to be less stable as explained in Section 5.6.1. To stabilise the problem a regularisation term should be used as explained in Section 5.2.3, a sufficient amount of receivers should be used and a sufficiently long time interval should be considered. Before experimenting with point-wise objectives, we suggest to consider an objective that assumes recordings on the entire boundary of the domain, i.e.,

$$J = \sum_{j=1}^{N_t} \frac{1}{2} \left\| \boldsymbol{v}_{\text{num}} \left(t_j \right) - \boldsymbol{v}_{\text{obs}} \left(t_j \right) \right\|_{\partial\Omega}^{2} + \mathcal{R}. \tag{5.46}$$

The second suggestions is on how to efficiently get an accurate result for the inverse problem. If the mesh used to solve the inverse problem is too coarse, it is quite useless to iterate the solution to high accuracy using the BFGS procedure. We suggest the following iterative procedure: Use a coarse mesh and a limited number of BFGS iterations to get a rough approximation of the unknown parameter. Next, project this solution on a finer mesh and use it as a bias in the regularisation term, and as initial value for solving the inverse problem on a finer mesh. As illustrated in Section 5.6.4, the value of the control after ten iterations using a proper Riesz map already holds valuable information.

5.7 Conclusions

We started this chapter by introducing typical inverse problems in NDT&E. We derived a classical deterministic formulation from a stochastic formulation, giving more insight in the problem formulation and highlighting the most important difficulties. Techniques for solving inverse problems were introduced and applied for determining wave speeds of a two dimensional elastic wave example. We compared generic gradient free methods and time reversal techniques with a gradient based optimisation strategy. The gradient based optimisation method proved to be the most successful method to tackle wave speed determinations. From the wave speed determination experiments we learned that gradient based techniques are the most sensible option to tackle inverse scattering problems, where spatially varying material parameters are to be retrieved. We showed that when using gradient based techniques for problems with spatially varying unknown parameters, sufficient attention has to be given to the function space in which these unknown controls are represented.

Chapter 6

Perfectly Matched Layers

6.1 Introduction

Three types of boundary conditions are frequently used in numerical wave propagation problems: reflecting boundaries, ports through which energy enters or leaves the system, and boundary conditions that mimic open space when truncating an infinite domain. A number of strategies for truncating infinite domains have been developed, including absorbing boundary conditions [48, 78], absorbing layers [54, 62] and one-way approximations [35, 74]. An absorbing layer introduces damping and is realised by extending the computational domain beyond the domain of interest. It is desirable to keep the size of the absorbing domain as small as possible to limit the additional computational work. However, none of the early damping layer techniques proved to be flawless.

In 1994, Bérenger [15] introduced an absorbing domain called Perfectly Matched Layers (PMLs). In a PML, waves are damped at a certain rate, described by an attenuation function (AF). It is desirable to use an 'optimal' AF in order to limit the size of the PML. Unfortunately, there is no universal recipe available to determine the best AF for specific problems. For particular cases, optimal PMLs can be found through mathematical analysis. For example, Chew and Jin [26] proved that for finite difference time domain methods, second-order polynomial AFs are optimal and suggested that these results should also be expected for finite element time domain methods. A generalisation of the analysis to more complicated cases (unstructured meshes, more general geometries and loads) is not straightforward, may be suboptimal or may even fail.

In this chapter we present an automatic calibration procedure for PMLs through optimisation of the PML parameters for a given problem. The functional we attempt to minimise is the energy left in the domain after an input signal should have left the domain of interest. The problem is constrained by the considered differential equation that describes the wave propagation of interest. We use gradient-based optimisation procedures to determine the parameters, with the adjoint of the forward problem used to compute the derivative of the target functional with respect to the PML parameters. We consider polynomial and piecewise-constant AFs, with the latter case motivating the introduction of what we will call 'Consecutive Matched Layers' (CMLs). An advantage of CMLs is that they are easily added to problems with arbitrary geometries, as we will show through numerical examples.

Numerical examples of the proposed procedure are presented for acoustic, elastodynamic and electromagnetic problems. The examples use the FEniCS/-DOLFIN [6, 67, 68] and dolfin-adjoint [38, 44] libraries. The complete source code to produce the presented examples is freely available and provided as supporting material [96].

The remainder of the chapter is organised as follows. Perfectly matched layers are introduced in Section 6.2. In Section 6.3, the formulation of Consecutive Matched Layers is presented, which is followed by the proposed procedure for automatic calibration of PMLs and CMLs in Section 6.4. We present and discuss test cases and results in Section 6.5. Conclusions are drawn in Section 6.6.

6.2 Perfectly matched layers

We denote the domain of physical interest by Ω_i, which is extended with an absorbing domain Ω_a ($\Omega_i \cap \Omega_a = \emptyset$), leading to the computational domain $\Omega = \Omega_i \cup \Omega_a$. To obtain a formulation for wave propagation problems with PMLs, we apply the technique of complex coordinate stretching [27, 91] to the generic wave equation in (2.7).

Solutions to wave equations are of the form

$$\boldsymbol{q}(\boldsymbol{x}, t) = \bar{\boldsymbol{q}}\left(\boldsymbol{x}\right) e^{-\imath\omega t}, \tag{6.1}$$

where $\bar{\boldsymbol{q}}(\boldsymbol{x})$ is the spatial solution, $\imath$ is the imaginary unit and ω is the angular frequency. A frequency domain formulation can be used by noting that $\dot{\boldsymbol{q}} = -\imath\omega\boldsymbol{q}$. PMLs in all directions are applied by introducing the coordinate transformations

$$\frac{\partial}{\partial x_i} \rightarrow \left(\frac{1}{1 + \imath\frac{\sigma_i(x_i)}{\omega}}\right)\frac{\partial}{\partial x_i}, \tag{6.2}$$

where $\sigma_i(x_i)$ are scalar attenuation functions (AFs) which are non-zero only in the absorbing region Ω_a and only depend on the component x_i of the spatial vector $\boldsymbol{x}$. The AFs will be defined at the end of the section.

To make the notation more compact, we introduce $\boldsymbol{\sigma}_i = \sigma_i(x_i)$ and denote combinations of different AFs in the index, e.g., $\boldsymbol{\sigma}_{ij+k} = \sigma_i(x_i)\sigma_j(x_j) + \sigma_k(x_k)$. Using (6.1) and applying the coordinate transformations in (6.2) to the wave equation (2.7) leads to

$$-\imath\omega\boldsymbol{q} + \sum_{i=1}^{3} \frac{\boldsymbol{F}_{i,i}}{1 - \frac{\boldsymbol{\sigma}_i}{\imath\omega}} = \boldsymbol{f}. \tag{6.3}$$

Multiplying (6.3) by all denominators appearing in it leads to

$$\begin{aligned} &-\imath\omega\left(\prod_{i=1}^{3}\left(1 - \frac{\boldsymbol{\sigma}_i}{\imath\omega}\right)\right)\boldsymbol{q} + \sum_{i=1}^{3}\left(\prod_{\substack{j=1\\ j\neq i}}^{3}\left(1 - \frac{\boldsymbol{\sigma}_j}{\imath\omega}\right)\right)\boldsymbol{F}_{i,i} \\ &\qquad = \left(\prod_{i=1}^{3}\left(1 - \frac{\boldsymbol{\sigma}_i}{\imath\omega}\right)\right)\boldsymbol{f}. \end{aligned} \tag{6.4}$$

With no source term inside the absorbing region Ω_a, we have $\boldsymbol{\sigma}_i\boldsymbol{f} = \boldsymbol{0}$ and the right-hand side of (6.4) simplifies to $\boldsymbol{f}$. Expanding the remaining products leads to

$$\begin{aligned} &-\imath\omega\boldsymbol{q} + \sum_{i=0}^{3}\boldsymbol{F}_{i,i} + \boldsymbol{\sigma}_{1+2+3}\boldsymbol{q} - \frac{1}{\imath\omega}\boldsymbol{\sigma}_{12+13+23}\boldsymbol{q} + \frac{1}{(\imath\omega)^2}\boldsymbol{\sigma}_{123}\boldsymbol{q} \\ &\qquad - \frac{1}{\imath\omega}\sum_{i=0}^{3}\left(\sum_{\substack{j=1\\ j\neq i}}^{d}\boldsymbol{\sigma}_j\right)\boldsymbol{F}_{i,i} + \frac{1}{(\imath\omega)^2}\sum_{i=0}^{3}\left(\prod_{\substack{j=1\\ j\neq i}}^{d}\boldsymbol{\sigma}_j\right)\boldsymbol{F}_{i,i} = \boldsymbol{f}. \end{aligned} \tag{6.5}$$

We return to a time domain formulation by deriving a system of first-order equations from (6.5) using two auxiliary fields, $\dot{\boldsymbol{r}} = \boldsymbol{q}$ and $\dot{\boldsymbol{s}} = \boldsymbol{r}$ and two auxiliary differential equations (ADEs) in addition to the wave equation:

$$\begin{aligned} \dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{F}_{2,2} + \boldsymbol{F}_{3,3} + \boldsymbol{\sigma}_{1+2+3}\boldsymbol{q} + \boldsymbol{r} &= \boldsymbol{f}, \\ \dot{\boldsymbol{r}} - \boldsymbol{\sigma}_{12+13+23}\boldsymbol{q} - \boldsymbol{\sigma}_{2+3}\boldsymbol{F}_{1,1} - \boldsymbol{\sigma}_{1+3}\boldsymbol{F}_{2,2} - \boldsymbol{\sigma}_{1+2}\boldsymbol{F}_{3,3} - \boldsymbol{s} &= \boldsymbol{0}, \\ \dot{\boldsymbol{s}} + \boldsymbol{\sigma}_{123}\boldsymbol{q} + \boldsymbol{\sigma}_{23}\boldsymbol{F}_{1,1} + \boldsymbol{\sigma}_{13}\boldsymbol{F}_{2,2} + \boldsymbol{\sigma}_{12}\boldsymbol{F}_{3,3} &= \boldsymbol{0}. \end{aligned} \tag{6.6}$$

In two spatial dimensions ($d = 2$) , we have the simplified system:

$$\begin{aligned} \dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{F}_{2,2} + \boldsymbol{\sigma}_{1+2}\boldsymbol{q} + \boldsymbol{r} &= \boldsymbol{f}, \\ \dot{\boldsymbol{r}} - \boldsymbol{\sigma}_2\boldsymbol{F}_{1,1} + \boldsymbol{\sigma}_1\boldsymbol{F}_{2,2} - \boldsymbol{\sigma}_{12}\boldsymbol{q} &= \boldsymbol{0}. \end{aligned} \tag{6.7}$$

In one spatial dimension ($d = 1$), there are no ADEs needed to describe the PML:

$$\dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{\sigma}_1\boldsymbol{q} = \boldsymbol{f}. \tag{6.8}$$

A specific PML is defined by the AFs $\boldsymbol{\sigma}_i$. The literature, e.g. [26], generally suggests polynomial AFs. For axis-aligned rectangular (cuboid) domains, polynomial AFs can be expressed as

$$\boldsymbol{\sigma}_i = \sigma_i(x_i) = \begin{cases} \displaystyle\sum_{j=0}^{n} c_{ij}\bar{x}_i^j & \text{if } x_i \in [a_{(i)0}, a_{(i)0} + w_i] \\ 0 & \text{otherwise,} \end{cases} \tag{6.9}$$

where n is the order of the polynomial, c_{ij} are the coefficients of the polynomial, $\bar{x}_i = g(x_i)$ is an affine transformation of x_i such that $g(x_i) = 0$ on the boundary between the domain of interest and the absorbing region, and $g(x_i) = 1$ on the exterior boundary of the absorbing region, $x_i = a_{(i)0}$ is the interface between Ω_i and Ω_a and w_i is the total width of the PML in the ith direction.

We also introduce a description of an AF with N piecewise-constant AFs of the form

$$\boldsymbol{\sigma}_i = \sigma_i(x_i) = \begin{cases} c_{ij} & \text{if } x_i \in [a_{(i)j}, a_{(i)j+1}] \ \forall j = 0 \dots N-1, \\ 0 & \text{otherwise,} \end{cases} \tag{6.10}$$

where $c_{ij} \geq 0$ are scalar values and $a_{(i)j} = a_i + j(w_i/N)$.

6.3 Consecutive matched layers

The complex coordinate stretching procedure used in the previous section results in the PML configuration depicted in Figure 6.1a. Overlapping PML regions lead to products of AFs that appear in the ADEs in (6.6). Solving for the auxiliary fields adds to the computational cost. To avoid this increase in cost, we adopt a simplification to the PML strategy.

When using non-overlapping absorbing domains, as depicted in Figure 6.1b, products of AFs are zero and (6.6) reduces to

$$\begin{aligned} &\dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{F}_{2,2} + \boldsymbol{F}_{3,3} + \boldsymbol{\sigma}_{1+2+3}\boldsymbol{q} + \boldsymbol{r} = \boldsymbol{f}, \\ &\dot{\boldsymbol{r}} - \boldsymbol{\sigma}_{2+3}\boldsymbol{F}_{1,1} - \boldsymbol{\sigma}_{1+3}\boldsymbol{F}_{2,2} - \boldsymbol{\sigma}_{1+2}\boldsymbol{F}_{3,3} = \boldsymbol{0}, \end{aligned} \tag{6.11}$$

which eliminates one ADE compared to (6.6). If we assume that $\boldsymbol{r} = \boldsymbol{0}$, which can be motivated by the fact that spatial derivatives in the second equation will be relatively small due to the damping, also the second ADE vanishes, further reducing (6.6) to

$$\dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{F}_{2,2} + \boldsymbol{F}_{3,3} + \boldsymbol{\sigma}_{1+2+3}\boldsymbol{q} = \boldsymbol{f}. \tag{6.12}$$

Since we prefer direction-independent AFs, we choose the AF in all directions to be defined by the same constant. Hence, $\boldsymbol{\sigma}_{1+2+3}$ can be replaced by an AF of the form

$$\sigma(\boldsymbol{x}) = \begin{cases} c & \text{if } \boldsymbol{x} \in \Omega_a \\ 0 & \text{otherwise,} \end{cases} \tag{6.13}$$

which allows to simplify (6.12) to

$$\dot{\boldsymbol{q}} + \boldsymbol{F}_{1,1} + \boldsymbol{F}_{2,2} + \boldsymbol{F}_{3,3} + \sigma(\boldsymbol{x})\boldsymbol{q} = \boldsymbol{f}. \tag{6.14}$$

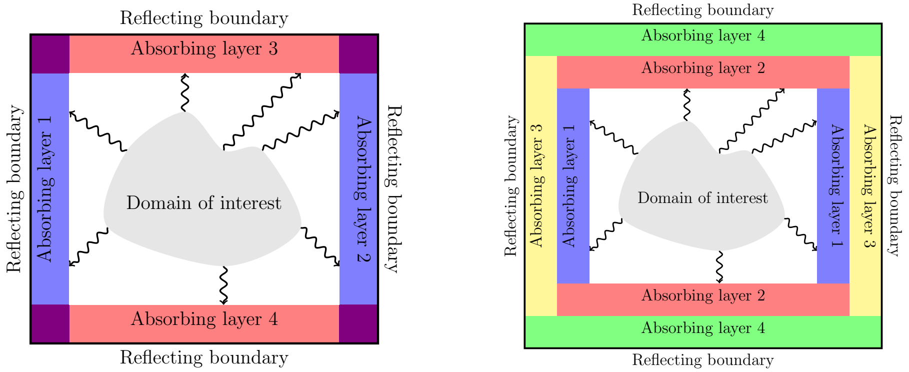
This formulation closely resembles the original absorbing layer strategy [54]. We however suggest to consider an absorbing domain Ω_a which is divided into N non-overlapping absorbing layers Ω_{a_j} such that $\Omega_a = \bigcup_{j=1}^{N} \Omega_{a_j}$, which leads to the following definition of the AF

$$\sigma(\boldsymbol{x}) = \begin{cases} c_j & \text{if } \boldsymbol{x} \in \Omega_{a_j} \\ 0 & \text{otherwise,} \end{cases} \tag{6.15}$$

where $c_j \geq 0$ is a constant scalar and Ω_{a_j} is the jth 'layer' of the absorbing domain.

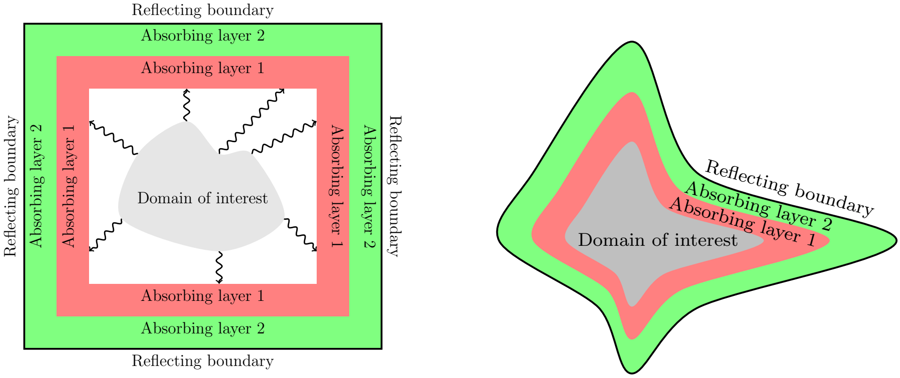
The resulting configuration is illustrated in Figure 6.1c, where the AF is constant on each colour/layer. Due to the absence of auxiliary fields, the computational cost is reduced relative to the PML model. A difficulty is how to choose the constants c_j that define the attenuation in the different layers. This issue will be addressed in the following section.

We note that a reasonable domain of interest can be extended with tightly wrapped layers, as shown in Figure 6.1d, and meshed conformingly. Hence this procedure can be applied to problems with arbitrary geometries, while avoiding complex mathematical interventions, e.g., as presented in [46].



(a) Adding perfectly matched layers in multiple directions to a geometry using complex coordinate stretching leads to overlapping regions.

(b) Multiple perfectly matched layers that have been added to a geometry without overlap.



(c) When two consecutive layers as shown in Figure 6.1b are defined by the same constant value, they can be considered as one merged layer. Fragmentation of a matched layer in this manner can be used to add tightly wrapped absorbing layers to an arbitrary geometry, as demonstrated in Figure 6.1d.

(d) Multiple tightly wrapped absorbing layers around an arbitrary geometry.

Figure 6.1: Graphical depiction of perfectly matched layers and consecutive matched layers.

By neglecting the ADE in (6.11) the absorbing domain is no longer a PML, hence we will refer to the simplified damping strategy as *consecutive matched layers* (CMLs). This name refers to the non-reflective free space boundary condition introduced in Katz et al. [62], mentioned as *absorbing layer* in Holland and Williams [54] and referred to as *matched layer* in Bérenger [15].

6.4 Automatic calibration of matched layer problems

The matched layer approaches presented in the preceding sections involved scalar AFs, $\sigma_i\left(x_i\right)$, and it is necessary to define their functional form. For finite difference methods, there are numerous papers describing how to determine the AFs, e.g, [13, 26, 29]. Chew and Jin [26] proved that quadratic polynomials result in optimal AFs for finite difference methods. Even if we presume that quadratic polynomial AFs are optimal for finite element time domain simulations, the question remains what precise form the quadratic polynomial should take for optimal results. Since users often want to add damping layers to their models without studying the truncation strategy in depth, an automatic determination procedure is appealing. We present a generic recipe for automatic calibration of the AF coefficients. The presented procedure is based on solving an optimisation problem.

6.4.1 Formulation of the optimisation problem

Our abstract optimisation problem is formulated as

$$\min_{\boldsymbol{y},\boldsymbol{u}} J(\boldsymbol{y},\boldsymbol{u}) \qquad \text{(objective functional)}, \tag{6.16}$$

$$\boldsymbol{c}(\boldsymbol{y},\boldsymbol{u}) = \boldsymbol{0} \qquad \text{(constraint)}, \tag{6.17}$$

where J is a scalar function, $\boldsymbol{y}$ is the state vector, $\boldsymbol{u} \in U_{\mathrm{ad}}$ is the control vector, U_{ad} is the set of the admissible values for the control values and $\boldsymbol{c}$ represents a set of constraints. For the considered problems, the state vector $\boldsymbol{y}$ contains $\boldsymbol{q}$ in the wave equation (2.7), and in the case of PML calibration is also contains the solutions of the auxiliary fields $\boldsymbol{r}$ and $\boldsymbol{s}$ in (6.6). The control vector $\boldsymbol{u}$ contains the c_{ij} parameters that define the AFs. Satisfaction of the constraint $\boldsymbol{c}(\boldsymbol{y},\boldsymbol{u})$ in our case is satisfaction of the wave equation with matched layers.

Remark 1. *The matched layer optimisation problem is in general not convex due to the non-linear relation between the controls and the states given by* *the constraint. This implies the likely existence of multiple local minima. Consequences of the existence of multiple local minima will be demonstrated by the numerical examples in Section 6.5. This prohibits us of considering the outcome of the automatic calibration procedure as optimal. Based on experimental results, we will however argue that the outcome is very likely to have near optimal performance.*

6.4.2 Measuring the quality of the absorbing region

To define an objective functional we need to quantify the quality of a matched layer. Typical quality measures involve the reflection coefficients, both at the interface between the domain of interest Ω_i and the matched layer Ω_a, and within the matched layer (see [26]). This *a priori* quality measure is difficult to manipulate in combination with finite element formulations.

We propose quantifying the quality of a matched layer through the amount of energy in a system at a judiciously chosen time for a judiciously chosen source term. With reflecting boundaries around the domain of interest and a vanishing input signal, the total energy in the system for the considered problems is constant once the input signal has vanished. If the domain of interest was embedded in an infinite domain, the total energy in the domain of interest would be zero at sufficiently large time. When absorbing layers are added to the domain of interest to mimic an infinite domain, the energy will reduce over time due to attenuation in the absorbing layers only, but it is highly unlikely that it will ever be exactly zero. The goal of the calibration procedure is to choose parameters for the matched layers such that the energy in the whole computational domain is minimised at a suitably chosen time, which we will call the *calibration time*, T_c. The reduction in energy in the numerical simulation at time T due to the absorbing layers is given by

$$\delta E = -10 \log_{10} \left(\frac{E\,(T)}{\bar{E}\,(T)} \right), \tag{6.18}$$

where $\bar{E}$, the energy in the whole computational domain with zero-valued AFs, is used as a reference value and E is the energy on the computational domain for the problem with non-zero AFs.

The energy in the system of an acoustic, electromagnetic and elastic wave problem was defined in Section 2.3.1. For the generic wave equation, the energy in the system is given by:

$$E(t) = \frac{1}{2} \left\langle \mathbf{Q}\boldsymbol{q}(t), \boldsymbol{q}(t) \right\rangle_{\Omega}, \tag{6.19}$$

where $\langle .,. \rangle_\Omega$ is the L^2 inner product over the entire computational domain Ω and $\mathbf{Q}$ is a matrix containing the material parameters.

The matrix $\mathbf{Q}$ in (6.19) for acoustic wave problems is

$$\mathbf{Q} = \operatorname{diag}\left(\rho,\ \rho,\ \rho,\ \frac{1}{K}\right), \tag{6.20}$$

where ρ is the mass density and K is the bulk modulus.

The matrix $\mathbf{Q}$ in (6.19) for electromagnetic wave problems is

$$\mathbf{Q} = \operatorname{diag}(\mu,\ \mu,\ \mu,\ \varepsilon,\ \varepsilon,\ \varepsilon), \tag{6.21}$$

where ε is the permittivity and μ is the permeability.

The matrix $\mathbf{Q}$ in (6.19) for elastic wave problems is

$$\mathbf{Q} = \operatorname{diag}\left(\rho,\ \rho,\ \rho,\ \boldsymbol{C}^{-1}\right), \tag{6.22}$$

where ρ is the mass density and $\mathbf{C}$ is the stiffness matrix as introduced in (2.15). Note that the matrix $\mathbf{Q}$ is block diagonal for this case.

6.4.3 Objective functional

The objective functional we use in calibrating matched layer problems is

$$J(\boldsymbol{q}, \boldsymbol{u}) = E(T_c), \tag{6.23}$$

where $E(T_c)$ is the energy in the system at the calibration time.

Another quantity of interest in designing matched layers is reflections at the interface between the domain of interest and the damping region. If the calibration time T_c is chosen too large, then energy can be damped gradually every time a wave encounters the damping region and is partially reflected by it. In order to include the effect of these reflections in J (6.23), the calibration time should be chosen such that reflections of the input signal at the material/matched layer interface encounter the damping region as few times as possible.

A practical concern is that the calibration time should be chosen as small as possible for computational speed, since a greater calibration time increases the number of time steps, and hence the cost of the optimisation process.

6.4.4 Computing gradients of the objective functional

We use gradient-based optimisation methods to calibrate the matched layer parameters. To compute the gradient of the objective functional J with respect to the control parameters $\boldsymbol{u}$, we use the adjoint approach [94]. In essence, we find $\mathrm{d}J/\mathrm{d}\boldsymbol{u}$ from

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{u}} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}} + \frac{\partial J}{\partial \boldsymbol{u}}, \tag{6.24}$$

where the adjoint variable $\boldsymbol{\lambda}$ is the solution to the system

$$\left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{q}}\right)^T \boldsymbol{\lambda} = -\left(\frac{\partial J}{\partial \boldsymbol{q}}\right)^T. \tag{6.25}$$

A detailed derivation for the time discretised problems can be found in Section 6.4.6. Key to the adjoint approach for computing derivatives of functionals is that that only one system needs to be solved to compute the gradient, regardless of the number of controls. Moreover, (6.25) is similar in structure to the system that is solved in the forward problem.

For the numerical examples in Section 6.5, in our implementations we express the forward model in FEniCS syntax [5, 67, 68], from which the adjoint problem is computed automatically by the library dolfin-adjoint [44].

6.4.5 Practical procedure

To automatically calibrate a PML or CMLs for a problem of interest we create a calibration set-up. The procedure is:

1. Extend the domain of interest with artificial layers Ω_a and mesh the entire domain with cell edges conforming to the boundary of Ω_i and Ω_a.
2. Extend the physical material parameters on the domain of interest to the absorbing region.
3. Set the attenuation in the damping region to zero.
4. Select an input signal with local support in time to fit the frequency range of the application under consideration.
5. Select a calibration time T_c, such that the peak of the input pulse has travelled at least once through the damping region in every direction at the lowest wave speed.
6. Update the AF parameters via a gradient-based optimisation process.

When the optimiser has converged, the obtained controls for the calibration set-up are used in the AF to solve the forward problem of interest. Note that the calibration set-up can differ from the problem of interest, as will be demonstrated for the electromagnetic example in Section 6.5. In the other example the geometry, mesh and excitation of the problem of interest and calibration set-up are kept. The final time of the problem of interest can differ from the calibration time, T_c. We will call the final time for the problem of interest the *evaluation time*, T_e.

6.4.6 Computing the gradient of the objective functional for the time discretised problem

Consider the generic wave equation with a PML in one spatial dimension as given in (6.8) with one constant matched layer added, discretised in time with the implicit trapezoidal rule:

$$\frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_n}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{q}'_{n+1} + \boldsymbol{q}'_n}{2} \right) + \sigma \frac{\boldsymbol{q}_{n+1} + \boldsymbol{q}_n}{2} = \frac{\boldsymbol{f}_{n+1} + \boldsymbol{f}_n}{2}, \tag{6.26}$$

where $\boldsymbol{q}_n$ is the computed approximation for $\boldsymbol{q}\,(n\Delta t)$, $\boldsymbol{f}_n$ is an evaluation of the source function $\boldsymbol{f}$ at time $n\Delta t$ and the accent indicates a spatial derivative, i.e., $\boldsymbol{q}'_n = \partial \boldsymbol{q}_n / \partial x_1$. We introduce the vector $\bar{\boldsymbol{q}} = \left(\boldsymbol{q}_1,\, \boldsymbol{q}_2,\, \ldots,\, \boldsymbol{q}_{n-1},\, \boldsymbol{q}_n\right)$ containing the solution at each time step. To be able to study the procedure in detail, we will restrict the time integration to computing three steps with a given initial value $\boldsymbol{q_0} = \mathbf{0}$ resulting in the states $\boldsymbol{q_1}$, $\boldsymbol{q_2}$ and $\boldsymbol{q_3}$ respectively. In that case the objective function is

$$J(\bar{\boldsymbol{q}}, \boldsymbol{u}) = \frac{1}{2} \left\langle \mathbf{Q}\boldsymbol{q}(t_3), \boldsymbol{q}(t_3) \right\rangle_{\Omega}, \tag{6.27}$$

and its derivative with respect to the state vector $\bar{\boldsymbol{q}}$ is

$$\frac{\partial J}{\partial \bar{\boldsymbol{q}}} = \begin{pmatrix} 0 & 0 & \mathbf{Q}\boldsymbol{q}_3 \end{pmatrix}. \tag{6.28}$$

The system of constraints in this case consists of three equations:

$$\begin{aligned}
\boldsymbol{c}_1 &= \frac{\boldsymbol{q}_1}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{q}'_1}{2} \right) + \sigma \frac{\boldsymbol{q}_1}{2} - \frac{\boldsymbol{f}_1 + \boldsymbol{f}_0}{2} = 0, \\
\boldsymbol{c}_2 &= \frac{\boldsymbol{q}_2 - \boldsymbol{q}_1}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{q}'_2 + \boldsymbol{q}'_1}{2} \right) + \sigma \frac{\boldsymbol{q}_2 + \boldsymbol{q}_1}{2} - \frac{\boldsymbol{f}_2 + \boldsymbol{f}_1}{2} = 0, \\
\boldsymbol{c}_3 &= \frac{\boldsymbol{q}_3 - \boldsymbol{q}_2}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{q}'_3 + \boldsymbol{q}'_2}{2} \right) + \sigma \frac{\boldsymbol{q}_3 + \boldsymbol{q}_2}{2} - \frac{\boldsymbol{f}_3 + \boldsymbol{f}_2}{2} = 0,
\end{aligned} \tag{6.29}$$

for which we can compute the Jacobian matrix

$$\frac{\partial \boldsymbol{c}}{\partial \bar{\boldsymbol{q}}} = \begin{pmatrix} \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}_1} & \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}_2} & \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}_3} \\ \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_1} & \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_2} & \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_3} \\ \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_1} & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_2} & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_3} \end{pmatrix} = \begin{pmatrix} \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}_1} & 0 & 0 \\ \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_1} & \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_2} & 0 \\ 0 & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_2} & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_3} \end{pmatrix}. \tag{6.30}$$

The partial derivatives in this matrix are computed from the following equations

$$\begin{aligned} \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{q}_i} &= \frac{\mathbf{I}}{\Delta t} + \mathbf{A}_1 \left(\frac{\mathbf{D}}{2} \right) + \sigma \frac{\mathbf{I}}{2}, \\ \frac{\partial \boldsymbol{c}_{\boldsymbol{i}}}{\partial \boldsymbol{q}_{i-1}} &= -\frac{\mathbf{I}}{\Delta t} + \mathbf{A}_1 \left(\frac{\mathbf{D}}{2} \right) + \sigma \frac{\mathbf{I}}{2}, \end{aligned} \tag{6.31}$$

where $\mathbf{I}$ is the identity matrix of size $d \times d$, with d the length of $\boldsymbol{q}_i$, and $\mathbf{D}$ is a $d \times d$ diagonal matrix with $\mathbf{D}_{ii} = \partial_{x_1}$. This information allows us to compute the adjoint states $\boldsymbol{\lambda}$ by solving

$$\left(\frac{\partial \boldsymbol{c}}{\partial \bar{\boldsymbol{q}}} \right)^T \boldsymbol{\lambda} = -\left(\frac{\partial J}{\partial \bar{\boldsymbol{q}}} \right)^T, \tag{6.32}$$

which in this case looks like

$$\begin{pmatrix} \frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{q}_1} & \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_1} & 0 \\ 0 & \frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{q}_2} & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_2} \\ 0 & 0 & \frac{\partial \boldsymbol{c}_3}{\partial \boldsymbol{q}_3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \mathbf{Q}\boldsymbol{q}_3 \end{pmatrix}. \tag{6.33}$$

Solving this system by back substitution leads to

$$\begin{aligned} 0 &= \frac{\boldsymbol{\lambda}_3}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{\lambda}_3'}{2} \right) + \sigma \frac{\boldsymbol{\lambda}_3}{2} + \mathbf{Q}\boldsymbol{q}_3, \\ 0 &= \frac{\boldsymbol{\lambda}_2 - \boldsymbol{\lambda}_3}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{\lambda}_2' + \boldsymbol{\lambda}_3'}{2} \right) + \sigma \frac{\boldsymbol{\lambda}_2 + \boldsymbol{\lambda}_3}{2}, \\ 0 &= \frac{\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2}{\Delta t} + \mathbf{A}_1 \left(\frac{\boldsymbol{\lambda}_1' + \boldsymbol{\lambda}_2'}{2} \right) + \sigma \frac{\boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2}{2}, \end{aligned} \tag{6.34}$$

which is identical to solving the forward problem (6.29) up to the variable names. In fact, by feeding $\boldsymbol{f}_0 = -2\mathbf{Q}\boldsymbol{q}_3$, $\boldsymbol{f}_1 = \boldsymbol{f}_2 = \boldsymbol{f}_3 = 0$ as a source to the forward solver, the adjoint states will be computed in reversed order. The gradient can now be computed by inserting the adjoint states in (5.21).

6.5 Numerical examples and discussion

We present examples using the calibration procedure for finite element acoustic, elastic and electromagnetic wave propagation problems, and consider both PMLs and CMLs. We start with a one-dimensional example, before moving on to two- and three-dimensional cases to examine performance with oblique incidence angles. We will consider PMLs for acoustic and elastodynamic examples, and CMLs for elastodynamic and electromagnetic examples. The computer code for reproducing all examples is available in the supporting material [96].

For all examples, we use the L-BFGS-B optimiser from SciPy [60]. This optimiser is a limited memory BFGS (see Section 5.4.1) implementation with bound support [24]. The bound support is used to prevent the optimiser choosing negative values for the piecewise-constant AFs. The optimiser stops when the gradient drops below a chosen threshold [77]. The threshold used in the different examples can be found in the supporting material [96].

To fully define the objective functional in (6.23), a calibration time and input signal have to be chosen. For all examples we use a Gaussian pulse. We choose the calibration time such that the peak of the input pulse has time to travel at least once to the boundary of the computational domain and back to the interface between the domain of interest and the absorbing domain at the lowest wave speed. Unless mentioned otherwise, first-order elements are used for all computations.

6.5.1 Perfectly matched layers

The examples presented in this section consider polynomial and piecewise-constant AFs for PMLs, as described in Section 6.2.

Acoustic wave propagation

We consider a rectangular domain of interest $\Omega_i = [0, 0.4]$ m $\times$ $[0, 0.1]$ m, which is extended at the right-hand boundary with a PML, as depicted in Figure 6.2. The domain is meshed with crossed-triangle cells with edge length 0.01 m in both x- and y-directions. Periodic boundary conditions are applied in the y-direction. An open boundary on the right-hand side of the domain is modelled by adding a PML in front of the reflecting boundary with $\boldsymbol{v} = \boldsymbol{0}$. On the left-hand boundary, the condition $\boldsymbol{v} = \left(\exp\left(-\left(4(t-t_0)/t_0\right)^2\right), 0\right)$ m/s is applied, where t_0 is the offset for the pulse. Note that for a large enough

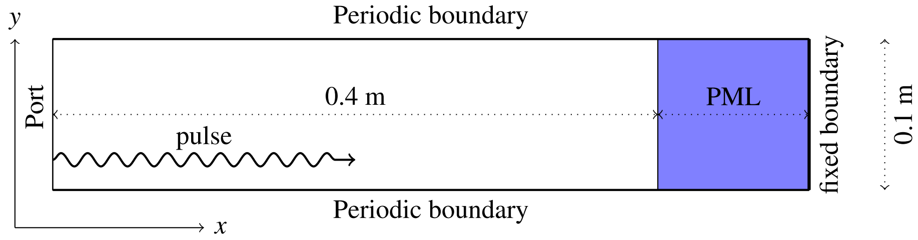
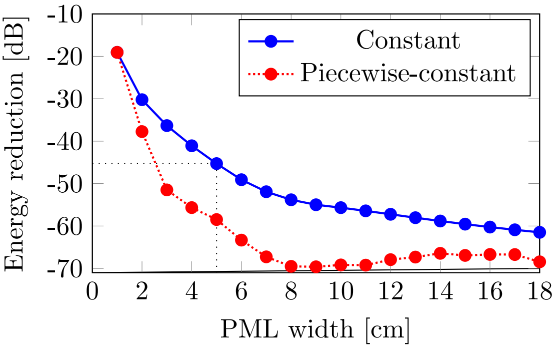


Figure 6.2: Geometry of the two-dimensional acoustic wave example with waves propagating in one direction.

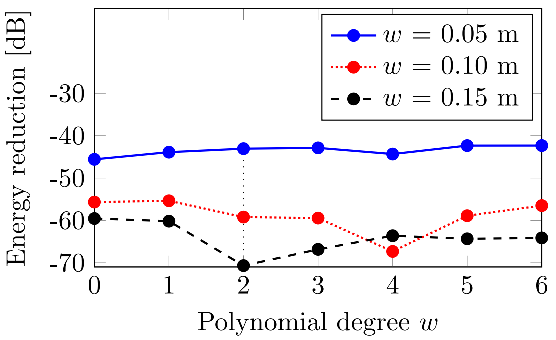
time $\boldsymbol{v}$ approaches zero and this boundary acts as a reflecting fixed boundary. We consider a homogeneous medium with mass density $\rho = 1.269$ kg/m^3 and bulk modulus $K = 101000$ Pa. The time step is 90% of the maximum time step allowed by the CFL condition, $\Delta t = 0.9 \times 0.01/(v\sqrt{2})$ s, where $v = \sqrt{K/\rho}$ is the wave speed for the medium. The offset of the pulse is chosen to be $t_0 = 100\Delta t$. The calibration time T_c is chosen to be the time the peak of the pulse needs to travel two and a half times through the domain of interest, $T_c = 2.5(0.4\text{m})v + t_0$. This way the peak of the pulse can encounter the PML interface only once, but there is sufficient time for the pulse to travel back-and-forth in the PML. For this example the calibration set-up is identical to the problem of interest, including the evaluation time $T_e = T_c$.

We first compare constant with piecewise-constant AFs for different PML widths. The smallest considered PML is 0.01 m wide. The PML is extended 0.01 m in x-direction seventeen times, up to a total width of 0.18 m. When a piecewise-constant AF is considered, one control value is added for every extension, e.g., for a 0.05 m wide PML, the piecewise-constant AF is defined by five control variables. The energy reduction, as defined in (6.18), for these experiments with the calibrated AFs is shown in Figure 6.3a. These results show that piecewise-constant AFs perform better than constant AFs for every PML width.

The reduction in energy for different polynomial order AFs and different PML widths is shown in Figure 6.3b. First note the results for the fourth-order polynomial AF, where the 0.10 m PML appears to outperform the 0.15 m PML. This peculiarity points to the optimisation problem being non-convex. We will comment on this further when examining initial guesses for the controls. Comparing Figure 6.3a and Figure 6.3b, it can be concluded that piecewise-constant AFs outperform the polynomial AFs, e.g., for a 0.10 m wide PML, the calibrated piecewise-constant AF reduces the energy more than any polynomial AF. We note from Figure 6.3b that there appears to be limited benefit in using polynomial orders greater than two, which is consistent with



(a) The solid blue curve shows the energy reduction for a constant attenuation function, and the dotted red curve for a piecewise-constant attenuation function as explained in Section 6.2.



(b) Perfectly matched layers are considered with three different widths w.

Figure 6.3: Energy reduction with perfectly matched layers for the acoustic wave example depending on the perfectly matched layer width (top) and polynomial degree for the attenuation function (bottom).

finite difference results presented by Chew and Jin [26]. We restrict further experiments to AFs to polynomial degrees of two or less.

We would expect the performance of a polynomial AF to be at least as good as the constant AF case since the polynomial case contains the constant case. However, Figure 6.3b shows that for a 0.05 m wide PML, a constant AF is slightly more effective than any other polynomial AF. This again points to the optimisation problem being non-convex.

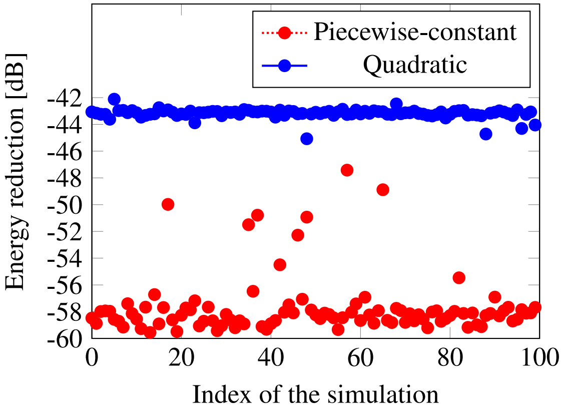


Figure 6.4: Energy reduction achieved by the calibrated attenuation functions for the acoustic wave example with zero initial values (index 0) and random sets of initial values (index > 0). The perfectly matched layer is 0.05 m wide. The experiment was performed for quadratic (dotted red) and piecewise-constant attenuation functions (solid blue).

We now fix the PML width to 0.05 m to examine the influence of the initial AF parameters. Figure 6.4 shows the reduction in energy after optimisation for zero initial values (index 0) and random starting values (indices greater than zero) for both a piecewise-constant and a quadratic AF. The starting values are uniformly sampled on the interval $[0, 7000]$ for the piecewise-constant case and the interval $[-500, 500]$ for the polynomial case. For the polynomial case, we allow negative coefficients in order to allow AFs that are not monotonically increasing. For the piecewise-constant AFs, the energy reduction for approximately ten percent of the results is more than 10 dB from the best result. There is less variation in the computed energy reduction for quadratic AFs compared to the piecewise-constant case. However, every piecewise-constant AF outperforms all quadratic AFs. In the remainder we will set the initial guess for all controls to zero.

Figure 6.5 shows the piecewise-constant AF for 0.02 m, 0.05 m, 0.10 m, 0.15 m and 0.19 m wide PMLs. The result is not immediately intuitive; the first control value is relatively large, followed by a substantially smaller second control value. The remaining control values are approximately equal and larger than the second value. The counter-intuitive outcome highlights an advantage of using an optimisation approach.

Figure 6.6 shows how the control values of the piecewise-constant AF change with each optimiser iteration for a 0.08 m wide PML together with the corresponding reduction in energy. For up to approximately 17 iterations the process favours

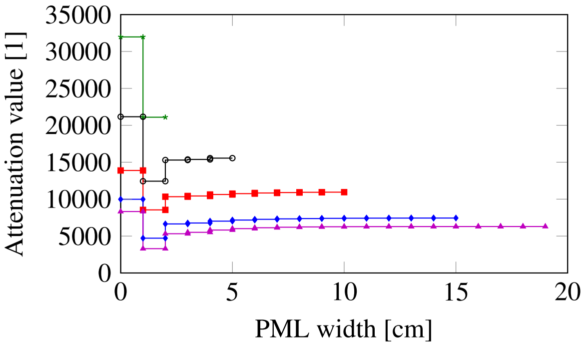


Figure 6.5: Optimal piecewise-constant attenuation functions for perfectly matched layers for the acoustic wave example for the case of 0.02 m (green stars), 0.05 m (black circles), 0.10 m (red squares), 0.15 m (blue diamonds) and 0.19 m (purple triangles) wide perfectly matched layer.

a constant AF. From the point at which the AF deviates significantly from a constant AF, a further 20dB to 30dB reduction in energy is observed.

Elastic wave propagation on a square

We simulate elastic wave propagation in an isotropic, homogeneous square domain $\Omega_i = [-6, 6]^2$ mm, which is extended in both the x- and y-directions with a 6 mm wide PML (see Figure 6.7a). We implement reflecting fixed boundaries on all sides of the computational domain. The longitudinal wave speed in the considered medium is $v_l = 5830.95$ m/s and the transverse wave speed $v_t = 3464.10$ m/s. The mass density of the considered material is 2500 kg/m^3. A Gaussian source $\boldsymbol{f} = (f_x, 0)$ is applied, where

$$f_x = \exp\left(-\left(\frac{t-50\Delta t}{50\Delta t/4}\right)^2\right)\exp\left(-\left(\frac{x}{10^{-6}}\right)^2\right)\exp\left(-\left(\frac{y}{10^{-6}}\right)^2\right). \quad (6.35)$$

A typical resulting elliptical wave front for this example is illustrated in Figure 6.7b.

The domain is meshed with crossed-triangle cells with edge length 1.2 mm. We solve this example using a discontinuous Galerkin finite element method, which is presented in Section 4.2. A time step size of $\Delta t = 4 \times 10^{-8}$ s is used. The calibration time is chosen to be $T_c = t_0 + (12 \text{ mm})\sqrt{2}/c_t$, which is the time needed for the peak to enter the domain and travel to the corner of the computational domain at the lowest wave speed. We consider the problem

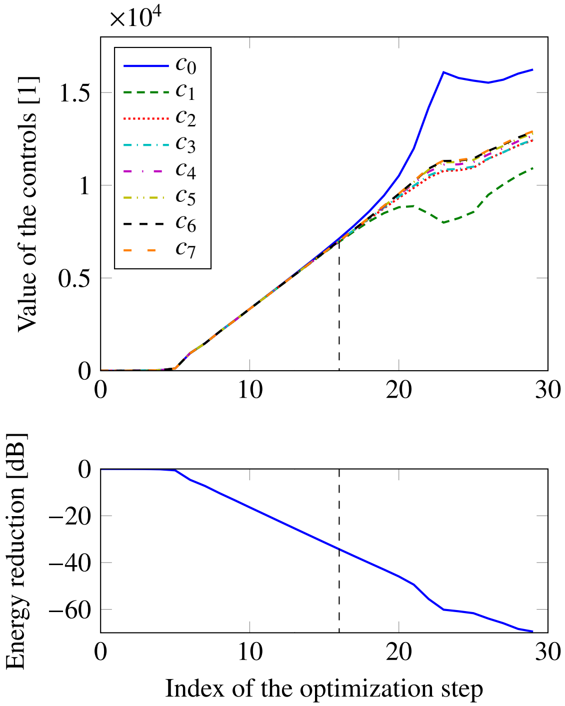
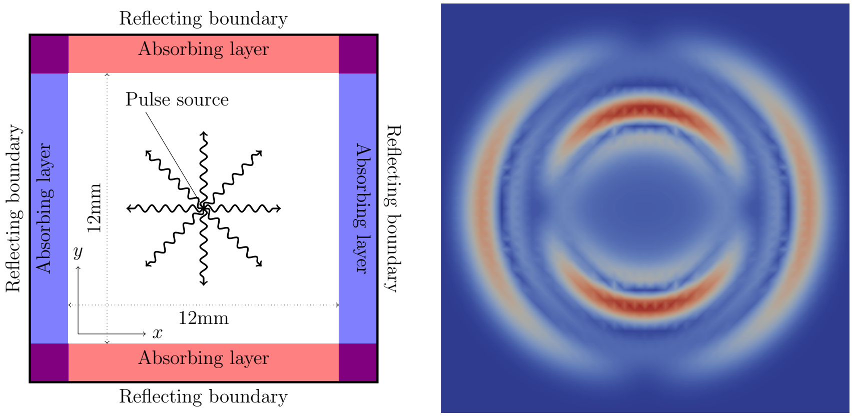


Figure 6.6: Evolution of controls (top) and reduction in energy (bottom) for the acoustic wave example as a function of the iteration step during the optimisation process for calibrating a piecewise-constant attenuation function with eight controls on a 0.08 m wide perfectly matched layer.

of interest to be identical to the calibration set-up with the exception of the evaluation time. Since the calibration time does not allow the wave to travel once in both direction through the absorbing layer, the reduction in energy would not include the full benefit of the absorbing layer. The evaluation time for this example is $T_e = t_0 + (24\text{ mm})\sqrt{2}/c_t$, which is the calibration time plus the time needed for the peak to travel back from the corner of the computational domain to the centre of the domain.

This model uses one AF in each spatial direction. It is however undesirable to have orientation dependent PMLs because of the symmetry of the domain. We therefore choose to define the AFs in both directions by the same control variables.

The evolution of the controls and reduction in energy at each optimiser step for



(a) (b)

Figure 6.7: Geometry of the elastic wave example with waves propagating in radial direction (left) with typical resulting wave propagation pattern (right).

the piecewise-constant case are shown in Figure 6.8. Despite the large changes in the control values at low iteration counts, the reduction in energy remains more-or-less constant from the second iteration. The final result is again not a monotonically increasing function, shown by the AF in Figure 6.9 (solid blue line).

As a second experiment for this model, we compare the piecewise-constant result to a quadratic AF. The evolution of the controls and reduction in energy during the optimisation process for a quadratic AF are shown in Figure 6.10. The calibration process for a quadratic AF has resulted in a constant AF (see Figure 6.9) and performs about 5% less well than the calibrated piecewise-constant AF.

6.5.2 Consecutive matched layers

We now move to examining the performance of the truncation strategy of consecutive matched layers presented in Section 6.3. In this case each ‘sub-layer’ has a constant attenuation function associated with it. The key difference with perfectly matched layers is the absence of auxiliary fields and equations in the model.

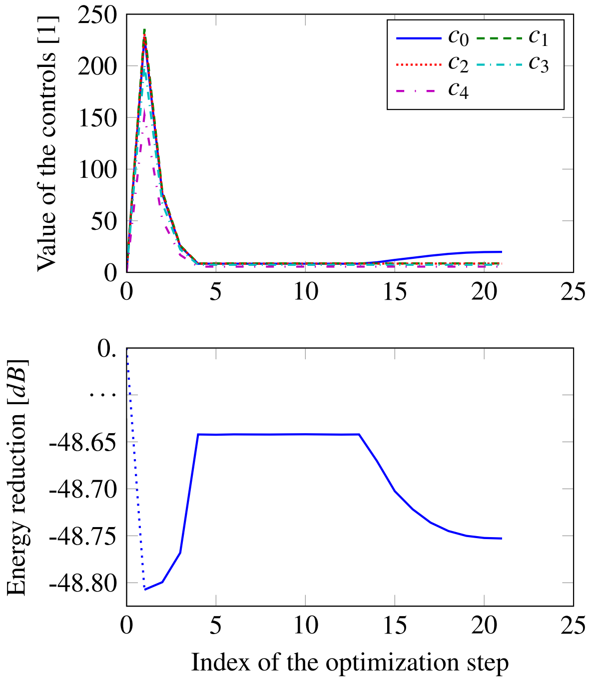


Figure 6.8: Evolution of the controls (top) and reduction in energy (bottom) for the elastic wave example as a function of the iteration step during the calibration process for a piecewise-constant attenuation function with five parameters for a 6 mm wide perfectly matched layer.

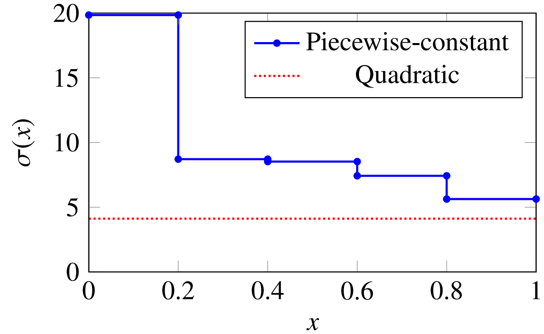


Figure 6.9: Piecewise-constant (solid blue) and quadratic (dotted red) attenuation function $\sigma(x)$ obtained with the calibration procedure for the elastic wave example with square geometry.

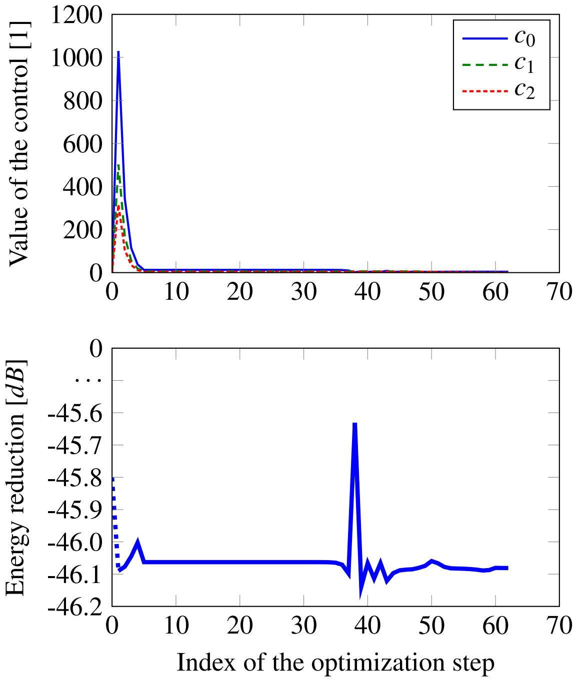


Figure 6.10: Evolution of the controls (top) and reduction in energy (bottom) for the elastic wave example with square geometry as a function of the iteration step during the optimisation process for calibrating a quadratic attenuation function for a 6 mm wide perfectly matched layer.

Elastic wave propagation on a square

We revisit the elastodynamic example from Section 6.5.1. Both the problem of interest and the calibration set-up are identical to the previous example with the PMLs replaced by CMLs. The evolution of the controls and reduction in energy during the optimisation process for calibrating the five CMLs are shown in Figure 6.11. The results shown in Figure 6.11 are almost identical to the PML results in Figure 6.8. Since the model for CMLs does not require ADEs, in contrast to the PML model, the automatic calibration procedure for CMLs is faster than for PMLs.

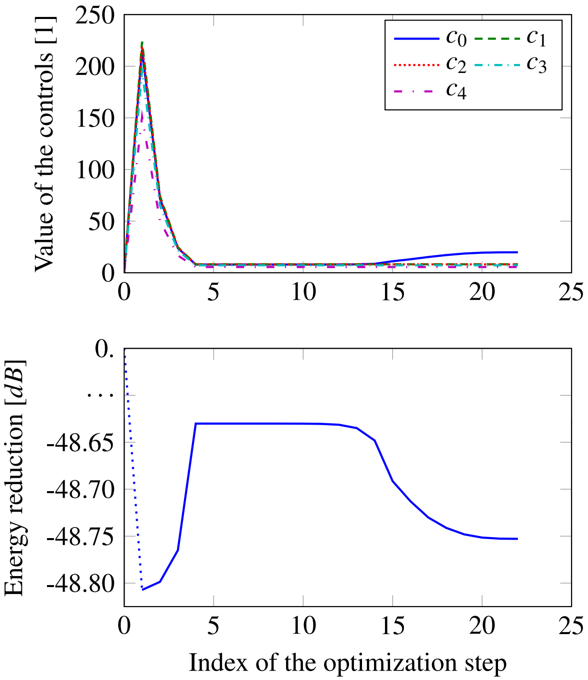


Figure 6.11: Evolution of the controls (top) and reduction in energy (bottom) for the elastic wave example as a function of the iteration step during the bounded calibration process for consecutive matched layers.

Elastic wave propagation on a more complicated geometry

We now adopt the problem of interest and calibration set-up of the preceding elastic example, but replace the square domain by the domain and mesh shown in Figure 6.12. The domain of interest is shown in dark blue. Five consecutive matched layers are placed around the domain of interest. The precise definition of the domain and the mesh are available in the supporting material [96].

The evolution of the attenuation constants for this problem and the corresponding reduction in energy are shown in Figure 6.13. The reduction in energy is only ten percent less than for the problem on the square domain. The attenuation in layers closer to the domain of interest is larger than for the layers farther from the domain of interest. This is probably a manifestation of the sensitivity of the different controls. It is to be expected that the attenuation in the outer layers has less effect on the reduction in energy, since a considerable

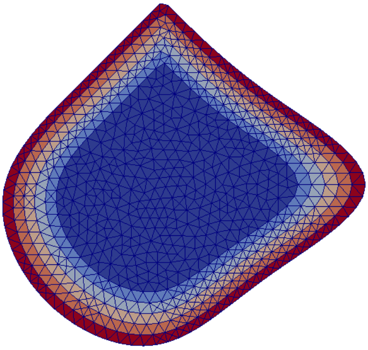


Figure 6.12: The more complicated computational domain used for the elastic wave experiment with consecutive matched layers. The domain of interest (dark blue) is surrounded by five tightly wrapped consecutive matched layers, each indicated by a distinct colour.

amount of energy will have been damped by layers closer to the domain of interest.

Electromagnetic wave propagation

We consider an application for which an absorbing region is calibrated, and then used to solve a problem of interest. The problem of interest involves a transverse electromagnetic wave [58] in a parallel plate wave guide. We solve equation (2.10) on the domain of interest $\Omega_i = [0, L_x] \times [0, L_y] \times [0, L_z(x)]$. We consider conducting plates at $x = 0$ and $x = L_x$, which are both modelled by implementing perfect electric conducting boundary conditions ($\boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{0}$) at $x = 0$ and $x = L_x$. The face at $z = 0$ is a port through which waves are inserted into the wave guide. We consider the case where the plates are infinite in y-direction, which is modelled by applying perfect magnetic conducting boundary conditions ($\boldsymbol{n} \times \boldsymbol{H} = \boldsymbol{0}$) at $y = 0$ and $y = L_y$. For $z \geq L_z(x)$ there is open space, which will be modelled using CMLs.

Before solving the problem of interest we calibrate the AFs on the adsorbing layer. To study the impact of oblique incidence angles at the boundary of the domain of interest, we will ‘stretch’ the upper conducting plate ($x = L_x$) in the z direction. Three configurations will be tested, i.e., with 90, 60 and 45 degree incidence angles. The domain with a 60 degree incidence angle is shown in Figure 6.14. The volume of absorbing layers will slightly differ in all three cases due to the different plate lengths, but the thickness of each layer (in the z-direction) is fixed.

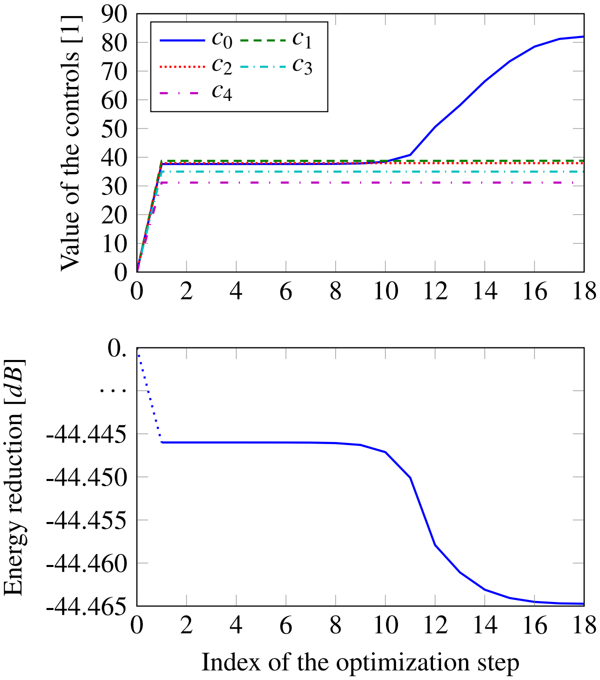


Figure 6.13: Evolution of the controls (top) and reduction in energy (bottom) for the elastic wave example with a complicated geometry as a function of the iteration step during the calibration process for consecutive matched layers.

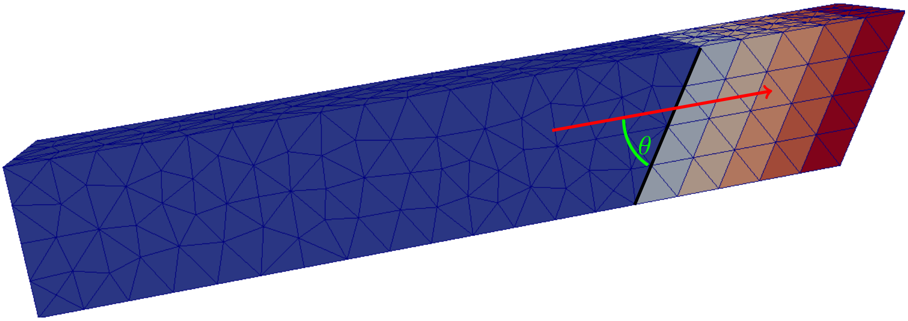


Figure 6.14: A parallel plate wave guide used for the electromagnetic example. The domain of interest (dark blue) is extended with five consecutive matched layers, each indicated by a different colour. In this problem, the waves enter the consecutive matched layers at a 60 degree angle θ. By modifying the length of the upper plate, the incidence angle can be controlled.

We extend the domain of interest with five absorbing layers, each one cell wide (see Figure 6.14). Also the boundary conditions of the domain of interest at $x = 0$, $x = L_x$, $y = 0$ and $y = L_y$ are extended to the absorbing domain. The boundary condition at the port ($z = 0$) is set to $\boldsymbol{E} = (E_x, 0, 0)$, where

$$E_x(t) = \exp\left(-\left(\frac{t - 10^{-8}}{10^{-8}/4}\right)^2\right). \tag{6.36}$$

At the end of the CMLs ($z = L_z(x)$) a perfect electric conducting boundary condition is applied. The calibration time is chosen to be the time for the peak of the input pulse to enter the system, move through the domain of interest, reflect off the interface between the domain of interest and the absorbing domain, and back to the source of the input signal, which is $T_c = 10^{-8} + 2L_z(0)/c$. To optimise the attenuation functions, we initialise the AFs to zero, and run the optimisation process for the 90, 60 and 45 degree incidence angle cases.

The evolution of the control variables and the corresponding reduction in energy at $T_e = T_c$ for the cases with 90, 60 and 45 degree incidence angles are shown in Figure 6.15, Figure 6.16 and Figure 6.17 respectively. We see that the two cases with non-perpendicular incidence perform well relative to the to the 90 degree case. The obtained attenuation values differ significantly between the three cases. The smaller the incidence angle, the more iterations are required to converge the optimisation algorithm.

We observe the least energy reduction for the 60 degree incidence case. The observation that the 45 degree incidence case performs better than both other cases is mainly because when a wave hits the interface between the domain of interest and the absorbing domain at a 45 degree incidence angle, the wave is reflected to the upper plate, hits it perpendicularly and hence is reflected again at a forty-five degree angle to the CMLs, before it gets reflected again in negative z-direction towards the source of the input signal. In other words, reflected waves meet the CMLs for a second time sooner than in the other cases.

To complete the electromagnetic wave case study, we compute a transverse electromagnetic wave in the wave guide with the 60 degree incidence angle and the CMLs that were calibrated for this case. The boundary conditions are as described for the calibration set-up, except now as an input wave we apply the boundary condition $\boldsymbol{E} = (E_x, 0, 0)$ at the port ($z = 0$), with

$$E_x(t) = \sin\left(3.1 \times 10^8 t\right). \tag{6.37}$$

There is no analytical solution available for a transverse electromagnetic wave in a parallel plate wave guide where one plate is longer than the other. However,

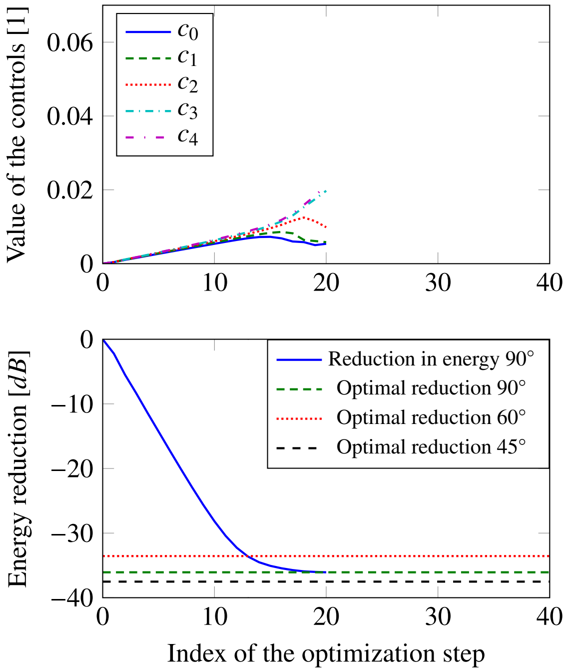


Figure 6.15: Evolution of the controls (top) and reduction in energy (bottom) for the electromagnetic wave example as a function of the iteration step during the calibration process for the consecutive matched layers for the domain with 90 degree incidence angle.

as the waves move from left to right in the wave guide, the solution in the rectangular part $\mathcal{R} = [0, L_x] \times [0, L_y] \times [0, L_z(0)]$ is not affected by the rest of the domain. Hence, we can use the analytical solution for a parallel wave guide with equal plates for $\mathcal{R}$ which is

$$\boldsymbol{E} = \left(\sin\left(3.1 \times 10^8 z/c - 3.1 \times 10^8 t\right), 0, 0\right),$$

$$\boldsymbol{H} = \left(0, \frac{1}{c\mu_0}\sin\left(3.1 \times 10^8 z/c - 3.1 \times 10^8 t\right), 0, 0\right), \quad (6.38)$$

where c is the speed of light and μ_0 is the permeability of vacuum.

For the initial condition, it is not straightforward to extend the analytical solution (6.38) into the absorbing region. Therefore, we start with a zero

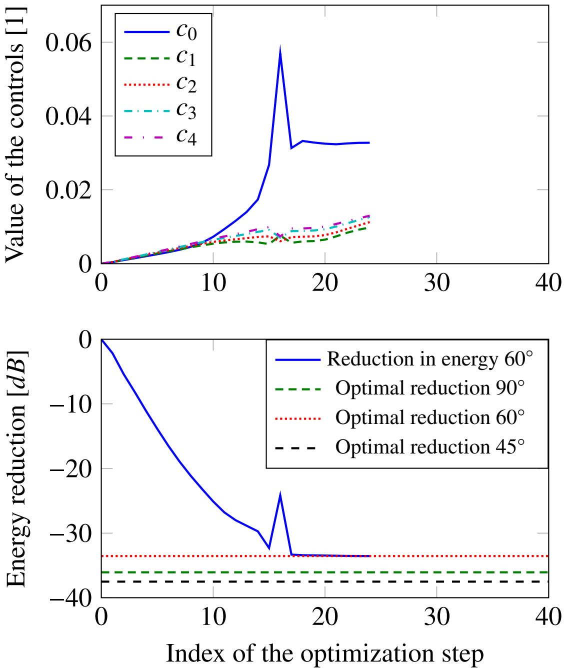


Figure 6.16: Evolution of the controls (top) and reduction in energy (bottom) for the electromagnetic wave example as a function of the iteration step during the calibration process for the consecutive matched layers for the domain with 60 degree incidence angle.

initial value and compare the numerical solution to the analytical solution after the problem reaches a steady state. To evaluate the numerical solution, we compare the electromagnetic energy (defined in (2.17)) of the numerical solution computed with third-order polynomial elements to the reference solution in (6.38) in $\mathcal{R}$ Figure 6.18. We see that, after reaching the steady state, the periods of the numerical and exact solutions are well aligned. Importantly, we see that there is no systematic increase in energy for the numerical case, which demonstrates that the CMLs are effective.

6.6 Conclusions

We have presented an approach to automatically calibrate attenuation functions for matched layers in wave propagation problems solved using finite element

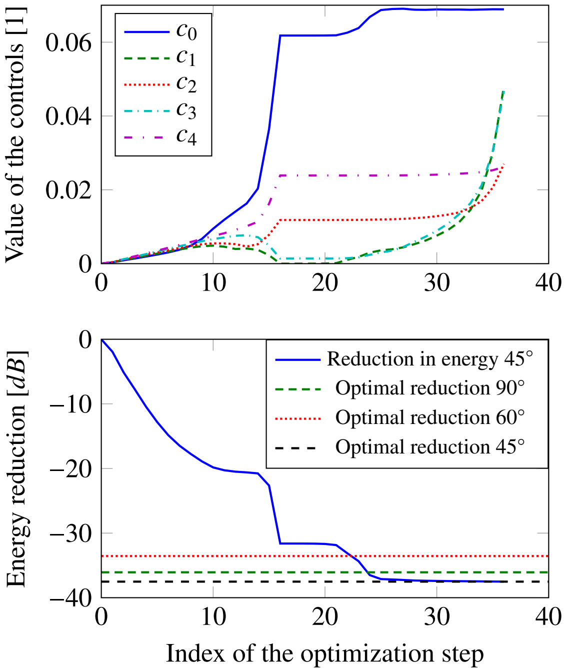


Figure 6.17: Evolution of the controls (top) and reduction in energy (bottom) for the electromagnetic wave example as a function of the iteration step during the calibration process for the consecutive matched layers for the domain with 45 degree incidence angle.

time domain methods. The presented procedure is not problem-specific, and in principle can be used to calibrate perfectly matched layers for any problem, regardless of the discretisation method. We have experimentally shown that there is no need to use polynomial attenuation functions higher than order two. Piecewise-constant attenuation functions can however result in equally effective perfectly matched layers. For piecewise-constant attenuation functions, the calibration procedure does not prefer monotonically increasing attenuation functions.

We have presented calibration of a damping strategy which we call *consecutive matched layers*. The automatic calibration procedure for consecutive matched layers is identical to the calibration procedure for perfectly matched layers. Consecutive matched layers lead to a simpler model than perfectly matched layers, resulting in shorter simulation times, for both the forward problem and the calibration procedure. It was shown for a collection of examples that

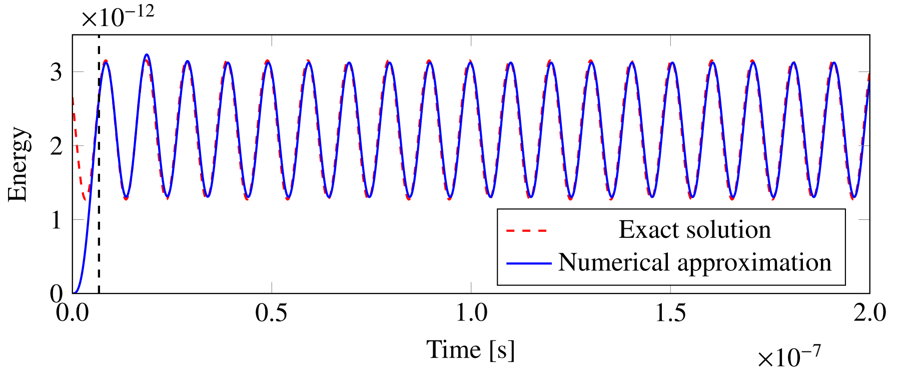


Figure 6.18: The computed electromagnetic energy (solid blue), as defined in (2.17) in a parallel plate wave guide with output port under a 60 degree angle compared to the theoretical reference (dashed red). The reference is only valid once a steady state has been reached.

consecutive matched layers can perform as well as perfectly matched layers. A major advantage of consecutive matched layers over perfectly matched layers is that consecutive matched layers can be easily applied to complex domains.

Chapter 7

Conclusion

The main goal of this work was to develop software for automated non-destructive testing and evaluation problems based on wave propagation phenomena. To numerically solve wave phenomena, the finite integration technique and the finite element method with continuous and discontinuous elements were studied. Non-destructive testing and evaluation problems were formulated and solved as inverse problems. Strategies for selecting and testing unknown material parameters were illustrated on a wave speed determination example. Also, an automatic calibration technique for absorbing layers, based on inverse solving techniques, was proposed and validated.

In this chapter we summarise the content of this work by going over the conclusions of the individual chapters on solving forward and inverse wave propagation problems and automatic calibration of absorbing layers. As the research in this work can be regarded as preliminary work to tackle more challenging problems in automated non-destructive testing and evaluation, indicating the range of applicability of this work and giving suggestions for future research are important notes to end with.

7.1 Solving forward wave propagation problems as part of solving inverse wave problems

In Chapter 3, the finite integration technique was studied to solve forward wave propagation problems on structured tensor-product meshes. For the purpose of simulating different wave propagation phenomena, a framework of eight staggered grids was presented. Using this framework, we have established the relation between mimetic and nodal discretisations of wave propagation problems. It was also shown that for the studied problems, it is time efficient to use higher order time integration methods when high accuracy is desired. If a low accuracy is sufficient, the classical leapfrog method is equally fast as higher order methods.

In Chapter 4, the finite element method with continuous and discontinuous elements was studied to solve forward wave propagation problems on unstructured meshes. We have harnessed the FEniCS/DOLFIN software to implement the studied problems. Special attention was given to elastic wave propagation, as we chose to impose symmetry on the numerical approximation of the stress tensor through the weak formulation when using continuous elements and through the function spaces when using discontinuous elements. This choice motivated a thorough efficiency study to compare both methods to solve elastic waves. From this study we have concluded that higher order time integration methods have no benefit over classical second order methods. When high accuracy is desired, discontinuous elements allow for faster computations. When low accuracy is sufficient, continuous elements are only slightly faster than discontinuous elements.

7.2 Automated non-destructive testing and evaluation

In Chapter 5, typical problems in non-destructive testing and evaluation were formulated as inverse problems. A general deterministic formulation was derived from a stochastic formulation to provide sufficient insight in inverse problems to correctly solve relevant problems. We discussed sampling, time reversal, gradient free and gradient based techniques to retrieve likely candidates for unknown material parameters by indirect observations. All mentioned techniques were illustrated on an elastic wave speed determination problem. We used a time of flight method to obtain a rough initial guess. Uniform sampling of controls was used in combination with a misfit test and in combination with a time reversal test to refine the initial guess. Also, a more sophisticated wave speed determination technique, purely based on time reversal, was considered as an alternative, but did not perform well for the considered example. The generic gradient free optimisation techniques of Nelder-Mead and Powell performed well, but are slow when used for problems with many unknown parameters. We showed that gradient based approaches were the most efficient option that was studied to retrieve wave speeds. The dolfin-adjoint library was used to automatically obtain the gradient information from the forward problem.

We argued that the suggested gradient based approach is the only viable option to study spatially dependent material parameters. To illustrate this, a gradient based approach was used to reconstruct a smoothly varying and discontinuously varying wave speed distribution. We showed that choosing a proper function space for the control benefits the performance of the method.

7.3 Automated calibration of absorbing layers

In Chapter 6, we have used gradient based optimisation methods to automatically calibrate absorbing layers. The main focus was on perfectly matched layers, but the procedure was also applied to simple consecutive absorbing layers. Using the procedure, we showed that there is no need to use polynomial attenuation functions higher than order two. A piecewise-constant attenuation function can result in equally effective perfectly matched layers. For piece-wise constant attenuation functions, the calibration procedure does not prefer monotonically increasing attenuation functions.

It was shown for a collection of examples that simple consecutive absorbing layers can perform equally well as perfectly matched layers. Simple consecutive absorbing layers have the advantage over perfectly matched layers that they can easily be applied to more complex domains.

7.4 Applicability of the presented work

The spatial discretisation and time integration methods in Chapters 3 and 4, the inverse solving techniques in Chapter 5 and the automatic calibration strategies in Chapter 6 can all be applied to acoustic, electromagnetic and elastic wave problems as presented. The introduced notation and tools allow a unified treatment of acoustic, electromagnetic and elastic waves in one, two and three dimensions. We illustrated how more classical formulations can be translated to our formulation. This results in the fact that the presented methods are applicable on a wide range of problems. The genericity could also be used as a means to translate problem specific techniques from one application field to another.

7.5 Suggestions for future research

The methods presented in this work allow for many aspects of inverse wave propagation problems to be studied. One example is to study the sensitivity of, e.g., time reversal refocussing quality with respect to wave speeds. As a second example, a sensitivity analysis of the effectiveness of absorbing layers with respect to the attenuation functions would provide valuable information to improve the automatic calibration process.

It would be useful to exploit the generic formulations in this work to transfer problem specific inverse problem solutions from, e.g., geophysics to accelerator physics. Another way to exploit the unified approach used in this work is to study multi-physics problems, such as air-coupled non-destructive testing set-ups.

As the suggested gradient based solution methods form a very general tool, they can be applied to a wide range of problems. As indicated in Section 5.6, more work needs to be done to construct a successful procedure. This work could greatly benefit from discussions with non-destructive testing and evaluation experts. A considerable amount of work and interaction will be needed to make this a solution method which is applicable to out-of-the-box problems as a true automated non-destructive testing and evaluation tool.

Solving inverse problems using gradient based approaches is computationally intensive. Further work is needed to speed up the process further. Many components of the solution process are however being researched by problem specialists. As this work has focussed on using existing tools to solve the problems of interest, we encourage researchers in the field of non-destructive testing and evaluation to focus on the application of these methods, as they will become faster and faster. It is however the opinion of the author that using gradient based optimisation techniques will in every case remain a computationally intensive procedure, which motivates to look even further. The presented methods could be used in mathematical programming with complementarity and equilibrium constraints to solve optimal sensor placement problems, e.g., for structural health monitoring [14]. Solving these kinds of problems would be even more computationally demanding, but the outcome could be deployed in many applications, reducing the overall cost of non-destructive testing and evaluation.

Bibliography

[1] COMSOL Multiphysics. http://www.comsol.com. pages 22

[2] deal.II. https://www.dealii.org/. pages 22

[3] J. D. Achenbach. *Wave propagation in elastic solids*. North-Holland Series in Applied Mathematics and Mechanics. North-Holland, 1973. pages 80

[4] S. Adams and B. Cockburn. A mixed finite element method for elasticity in three dimensions. *Journal of Scientific Computing*, 25:512–521, 2005. pages 60

[5] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans Math Software*, 40(2):9:1–9:37, 2014. pages 49, 130

[6] M. S. Alnæs, J. Blechta, J. Hake, J. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100):9–23, 2015. pages 22, 49, 91, 122

[7] D. N. Arnold and J. J. Lee. Mixed finite element methods for linear elasticity with weakly imposed symmetry. *SIAM Journal of Numerical Analysis*, 52(6):2743–2769, 2014. pages 61, 68

[8] D. N. Arnold and A Logg. Periodic table of the finite elements. *SIAM News*, 47(9), 2014. pages 49

[9] D. N. Arnold and R. Winther. Mixed finite elements for elasticity. *Numer. Math.*, 92:401 – 419, 2002. pages 60

[10] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39(5):1749 – 1779, 2002. pages 55

[11] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 76:1–155, 2006. pages 57

[12] D. N. Arnold, R. Falk, and R. Winther. Mixed finite element methods for linear elasticity with weakly imposed symmetry. *Mathematics of Computation*, 15(260):1699 – 1723, 2007. pages 61

[13] S. Asvadurov, V. Druskin, M. Guddati, and L. Knizhnerman. On optimal finite-difference approximation of PML. *SIAM Journal on Numerical Analysis*, 41(1):287–305, 2003. pages 127

[14] D. Balageas, C-P Fritzen, and A Güemes. *Structural Health Monitoring.* Wiley, 2006. pages 154

[15] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185 – 200, 1994. pages 121, 127

[16] M. Bihn. *Zur numerischen Berechnung elastischer Wellen im Zeitbereich.* PhD thesis, TU Darmstadt, Germany, 1998. pages 26, 27, 33, 34

[17] A. Borzi and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations.* SIAM, Philadelphia, 2012. pages 112

[18] A. Bossavit. *Computational Electromagnetism.* Academic Press, San Diego, 1998. pages 33, 34, 58

[19] A. Bossavit. The discrete Hodge operator in electromagnetic wave propagation problems. In A. Bermúdez, D. Gómez, C. Hazard, P. Joly, and J.E. Roberts, editors, *Proceedings of the Fifth International Conference on Mathematical and Numerical Aspects of Wave Propagation*, pages 753–759, Philadelphia, 2000. SIAM. pages 34

[20] A. Bossavit and L. Kettunen. Correction to “Yee-like schemes on staggered cellular grids: a synthesis between FIT and FEM approaches". *IEEE Transactions on Magnetics*, 36(6):4050, nov 2000. pages 25

[21] M. A. Botchev and J. G. Verwer. Numerical integration of damped Maxwell equations. *SIAM Journal Scientific Computing*, Vol. 31, No. 2: 1322–1346, 2008. pages 37, 40

[22] O. Bou Matar, Y. Li, and K. Van Den Abeele. On the use of a chaotic cavity transducer in nonlinear elastic imaging. *Applied Physics Letters*, 95(141913), 2009. pages 88

[23] O. Bou Matar, P.-Y. Guerder, Y. Li, B. Vandewoestyne, and K. Van Den Abeele. A nodal discontinuous Galerkin finite element method for nonlinear elastic wave propagation. *The Journal of the Acoustical Society of America*, 131(5):3650–3663, 2012. pages 56

[24] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16, 5:1190–1208, 1995. pages 133

[25] F. Campia and M. Meo. Acoustic emission localization in complex dissipative anisotropic structures using a one-channel reciprocal time reversal method. *The Journal of the Acoustical Society of America*, 130: 168–175, 2011. pages 87

[26] W. C. Chew and J. M. Jin. Perfectly matched layers in the discretized space: An analysis and optimization. *Electromagnetics*, 16(4):325–340, 1996. pages 121, 124, 127, 128, 135

[27] W. C. Chew and W. H. Weedon. A 3-d perfectly matched medium from modified Maxwell's equations with stretched coordinates. *Micro. Opt. Tech. Lett.*, 7(13):599–604, 1994. pages 122

[28] M. Clemens and T. Weiland. Discrete electromagnetism with the finite integration technique. *Progress In Electromagnetic Research*, PIER 32: 65–87, 2001. pages 25

[29] F. Collino and P. B. Monk. Optimizing the perfectly matched layer. *Computer Methods in Applied Mechanics and Engineering*, 164(1–2):157 – 171, 1998. pages 127

[30] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Applied Mathematical Sciences. Springer Berlin Heidelberg, second edition, 1997. pages 80

[31] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, Vol. 100, No. 1:32–74, 1928. pages 19, 44

[32] CST - Computer Simulation Technology. CST MICROWAVE STUDIO. http://www.cst.com. pages 38, 43

[33] C. Draeger and Fink M. One-channel time reversal of elastic waves in a chaotic 2d-silicon cavity. *Physical Review Letters*, 79(3):407–410, 1997. pages 87, 88

[34] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers : with applications in incompressible fluid dynamics.* Oxford University Press, Oxford, 2014. pages 68

[35] B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74 (5):1765–1766, 1977. pages 121

[36] P. Farrell and S. Funke. libadjoint. https://bitbucket.org/dolfin-adjoint/libadjoint. pages 22

[37] P. E. Farrell and S. W. Funke. Personal communication, 2016. pages 111

[38] P. E. Farrell, D. A. Ham, S. W. Funke, and M. E. Rognes. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing*, 35(4):C369–C393, 2013. pages 22, 91, 104, 122

[39] P. Fellinger, R. Marklein, K. J. Langenberg, and S. Klaholz. Numerical modeling of elastic wave propagation and scattering with EFIT-elastodynamic finite integration technique. *Wave Motion*, vol:21 iss:1: 47–66, 1995. pages 26

[40] M. Fink. Time-reverse acoustics. *Scientific American*, 281(5):91–97, 1999. pages 87

[41] M. Fink. Time-reversal waves and super resolution. *Journal of Physics: Conference Series*, 124:012004, 2008. pages 88

[42] M. Fink and Rosny J. Time-reversed acoustics in random media and in chaotic cavities. *Nonlinearity*, 15(1), 2002. pages 88

[43] S. Funke. Moola. https://github.com/funsim/moola/. pages 112

[44] S. W. Funke and P. E. Farrell. A framework for automated PDE-constrained optimisation. *submitted*, 2013. arXiv:1302.3894 [cs.MS]. pages 22, 91, 104, 122, 130

[45] S. W. Funke and M. Nordaas. PDE-constrained optimisation in hilbert spaces. Oral presentation during FEniCS 2014 workshop in Paris, France, 2014. pages 111

[46] H. Gao and J. Zhang. Implementation of perfectly matched layers in an arbitrary geometrical boundary for elastic wave modelling. *Geophysical Journal International*, 174(3):1029–1036, 2008. pages 125

[47] J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations.* Dover Publications, New York, 1952. pages 83

[48] T. Hagstrom. Radiation boundary conditions for the numerical simulation of waves. *Acta Numerica*, 8:47–106, 1999. pages 121

[49] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*, volume 8 of *Springer Series in Computational Mathematics*. Springer, Berlin, second revised edition, 2000. pages 36, 46, 63

[50] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer, Berlin, second edition, 2010. pages 36, 37, 38

[51] J. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, Berlin, 2008. pages 18, 53, 54, 56, 66

[52] R. Hiptmair. Discrete Hodge operators. *Numerische Mathematik*, 90: 265–289, 2001. pages 34

[53] W. V. D. Hodge. *The Theory and Applications of Harmonic Integrals*. Cambridge U.P., Cambridge, 1941. pages 58

[54] R. Holland and J. W. Williams. Total-field versus scattered-field finite-difference codes: A comparative assessment. *Nuclear Science, IEEE Transactions on*, 30(6):4583–4588, Dec 1983. pages 121, 125, 127

[55] J. M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *Computers and Mathematics with Applications*, 33(4):81 – 104, 1997. pages 25, 33

[56] J. M. Hyman and M. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Applied Numerical Mathematics*, 25(4):413 – 442, 1997. pages 33, 34

[57] J. M. Hyman and M. Shashkov. Mimetic discretizations for Maxwell's equations. *Journal of Computational Physics*, 151:881 – 909, 1999. pages 25

[58] J. D. Jackson. *Classical Electrodynamics*. Wiley, United States, 1998. pages 15, 143

[59] E. Janssen and K. Van Den Abeele. Exploiting time reversal principles for the accurate determination of material velocities. In A. S. Paipetis, T. E. Matikas, D. G. Aggelis, and D. Van Hemelrijck, editors, *Proc. 5th conference on Emerging Technologies in NDT*, pages 245–251, 2011. pages 88, 100, 101, 102, 107

[60] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed 2014-12-02]. pages 133

[61] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems.* Springer-Verlag, New York, 2005. pages 81, 82, 85, 92

[62] I. Katz, D. Parks, A. Wilson, M. Rotenberg, and J. Harren. Non-reflective free space boundary conditions for SGEMP codes. *Systems, Science and Software*, SSS-R-76-2934, May 1976. pages 121, 127

[63] K. Kennedy and K. S. McKinley. Optimizing for parallelism and data locality. In *ACM International Conference on Supercomputing 25th Anniversary Volume*, pages 151–162, New York, NY, USA, 2014. ACM. pages 66

[64] R. C. Kirby. From functional analysis to iterative methods. *SIAM Rev.*, 52(2), January 2010. pages 111

[65] J. Krautkrämer and H. Krautkrämer. *Ultrasonic Testing of Materials.* Springer-Verlag Berlin Heidelberg, 1990. pages 7, 12, 96

[66] C. Larmat, J. Tromp, Q. Liu, and J. P. Montagner. Time reversal localisation of glacial earthquakes. *Journal of Geophysical Research*, 113 (B9), 2008. pages 88

[67] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Trans Math Software*, 37(2):20:1–20:28, 2010. pages 22, 49, 91, 122, 130

[68] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering.* Springer, 2012. pages 22, 49, 59, 91, 122, 130

[69] K.-A. Mardel and R. Winther. Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications*, 18(1):1–40, April 2011. pages 111

[70] R. Marklein. *Numerische Verfahren zur Modellierung von akustischen, elektromagnetischen, elastischen und piezoelektrischen Wellenausbreitungsproblemen im Zeitbereich basierend auf der Finiten Integrationstechnik.* PhD thesis, Universität Gesamthochschule Kassel, Germany, 1997. pages 15, 26, 28, 33, 34

[71] MathWorks. MATLAB. http://www.mathworks.com/products/matlab/. pages 22

[72] P. B. Monk and A. K. Parrott. A dispersion analysis of finite element methods for Maxwell's equations. *SIAM Journal of Scientific Computing*, 15:916–937, 1994. pages 17

[73] K.W. Morton and D. F. Mayers. *Numerical solution of partial differential equations.* Cambridge University Press, Cambridge, 2005. pages 20

[74] G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *Electromagnetic Compatibility, IEEE Transactions on*, EMC-23(4):377–382, Nov 1981. pages 121

[75] J. C. Nédélec. Mixed finite-elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35 (3):315–341, 1980. pages 60

[76] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. pages 87

[77] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer, 233 Spring Street, New York, NY 10013, USA, second edition, 2000. pages 89, 133

[78] A. F. Peterson. Absorbing boundary conditions for the vector wave equation. *Microwave and Optical Technology Letters*, 1(2):62–64, 1988. pages 121

[79] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964. pages 87

[80] C. Prada, E. Kerbat, D. Cassereau, and M. Fink. Time reversal techniques in ultrasonic nondestructive testing of scattering media. *Inverse Problems*, 18(6):1761, 2002. pages 88

[81] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2008. pages 68

[82] P. A. Raviart and J. M. Thomas. Primal hybrid finite-element methods for 2nd-order elliptic equations. *Mathematics of Computation*, 31(138): 391–413, 1977. pages 58, 60

[83] W. Rudin. *Functional Analysis.* McGraw-Hill, Boston, second edition, 1991. pages 111

[84] E. H. Saenger, N. Gold, and S. A. Shapiro. Modeling the propagation of elastic waves using a modified finite-difference grid. *Wave Motion*, 31(1): 77 – 92, 2000. pages 35

[85] D. Sármány, M. A. Botchev, and J. J. W. van der Vegt. Dispersion and dissipation error in high-order runge-kutta discontinuous galerkin discretisations of the maxwell equations. *Journal of Scientific Computing*, 33(1):47–74, 2007. pages 19

[86] M. Scalerandi, M. Griffa, and P.A. Johnson. Robustness of computational time reversal imaging in media with elastic constant uncertainties. *Journal of Applied Physics*, 106(114911), 2009. pages 91, 96

[87] R. Schuhmann and T. Weiland. Conservation of discrete energy and related laws in the finite integration technique. *Progress In Electromagnetics Research*, PIER 32:301–316, 2001. pages 33, 36

[88] A. M. Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, May 2010. pages 82

[89] A. Taflove and S. C. Hagness. *Computational Electrodynamics: The finite difference time domain method.* Artech House, Norwood, 2005. Third edition. pages 36

[90] F. L. Teixeira and W. C. Chew. Lattice electromagnetic theory from a topological viewpoint. *Journal of Mathematical Physics*, Vol. 40, No. 1: 169–187, 1999. pages 34

[91] F. L. Teixeira and W. C. Chew. Complex space approach to perfectly matched layers: a review and some new developments. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 13(5):441–455, 2000. pages 122

[92] A. N. Tikhonov and V. T. Arsenin. *Solutions of Ill-Posed Problems.* V. H. Winston & Sons, Washington, 1977. pages 84

[93] E. Tonti. On the geometrical structure of electromagnetism. *Gravitation, Electromagnetism and Geometrical Structures*, Vol. 18, No 1:281–308, 1996. pages 34

[94] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications.* American Mathematical society, Providence, Rhode Island, 2010. Graduate Studies in Mathematics, Volume 112. pages 130

[95] B. Van Damme, K. Van Den Abeele, and O. Bou Matar. The vibration dipole: A time reversed acoustics scheme for the experimental localisation of surface breaking cracks. *Applied Physics Letters*, 100(8):084103–084103, 2012. pages 44

[96] S. Vandekerckhove. Automatic calibration of damping layers in finite element time domain simulations, January 2016. URL `http://dx.doi.org/10.5281/zenodo.45296`. pages 105, 122, 133, 142

[97] J. G. Verwer and M. A. Botchev. Unconditionally stable integration of Maxwell's equations. *Linear Algebra and its Applications*, Vol. 431, No. 431:300–317, 2009. pages 36, 37

[98] T. Weiland. Eine Methode zur Lösung der Maxwellschen Gleichungen für sechskomponentige Felder auf diskreter Basis. *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, 31(3):116–120, 1977. pages 26, 33

[99] T. Weiland. Time domain electromagnetic field computation with finite difference methods. *Int. J. Numer. Model.*, Vol. 9, No. 4:295–319, 1996. pages 25

[100] H. Whitney. *Geometric Integration Theory*. Princeton U.P., Princeton, 1957. pages 58

[101] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, Vol. 14:302–307, 1966. pages 36

Index

Curriculum vitae

Personal data

Name: Steven Vandekerckhove
E-mail: steven@vdks.be
Adress: Oudenaardsesteenweg 47, 8500 Kortrijk, Belgium
Nationality: Belgian
Date of birth: March 1988
Place of birth: Kortrijk, Belgium

Education

- **2000-2006** Science-Mathematics, Don Boscocollege, Kortrijk, Belgium
- **2006-2008** Bachelor in mathematics, KU Leuven Kulak, Belgium
- **2008-2009** Bachelor in mathematics, KU Leuven, Belgium
- **2009-2011** Master in engineering: mathematical engineering, KU Leuven, Belgium
  Master thesis (Dutch): *Tijdstapmethoden voor de 3D Maxwell golfvergelijking gediscretiseerd met Whitney elementen* Promotors: H. De Gersem and S. Vandewalle
- **2011-2016** Doctor in Physics: Wave propagation and signal processing research group, KU Leuven Kulak, Belgium

List of publications

International publications

S. Vandekerckhove, B. Vandewoestyne, H. De Gersem, K. Van Den Abeele and S. Vandewalle. Mimetic discretisation and higher order time integration for acoustic, electromagnetic and elastodynamic wave propagation. *Journal of Computational and Applied Mathematics*, 259, Part A, 65-76, 2014.

S. Vandekerckhove, G. N. Wells, H. De Gersem and K. Van Den Abeele. Automatic calibration of damping layers in finite element time domain simulations. *In preparation.*

Communications at international conferences, symposia and workshops

S. Vandekerckhove, H. De Gersem and S. Vandewalle (2011). WETD Simulation of the damped Maxwell wave equation. Oral presentation at *Workshop on Advances in Electromagnetic Research 2011.* Riezlern, Austria, 20 - 26 August 2011.

S. Vandekerckhove. Higher order time integration of wave equations. Oral presentation at *Innovative Time Integration.* Innsbruck, Austria, 13 - 16 May 2012.

S. Vandekerckhove, B. Vandewoestyne, H. De Gersem, S. Vandewalle and K. Van Den Abeele. Simulation of acoustic, electromagnetic and elastodynamic wave equations using the finite integration technique. Oral presentation at *International Congress on Computational and Applied Mathematics.* Gent, Belgium, 9 - 13 July 2012.

S. Vandekerckhove, H. De Gersem and S. Vandewalle. Time integration of the 3D Maxwell wave equation. Oral presentation at *European Conference on Mathematics for Industry 2012.* Lund, Sweden, 23 - 27 July 2012.

S. Vandekerckhove, B. Vandewoestyne, H. De Gersem, S. Vandewalle and K. Van Den Abeele. Application of the finite integration technique to solve the elastodynamic wave equation. Poster presentation at *Workshop on Advances in Electromagnetic Research 2012.* Riezlern, Austria, 1 - 7 October 2012.

S. Vandekerckhove, H. De Gersem and G. N. Wells. Solving electromagnetic wave propagation problems using FEniCS. Oral presentation at *Workshop on Advances in Electromagnetic Research 2013.* Riezlern, Austria, 17 - 23 August 2013.

S. Vandekerckhove. Optimization of perfectly matched layers in wave propagation problems. Poster presentation at *Thirty-eighth Conference of the Dutch-Flemish Numerical Analysis Communities.* Zeist, The Netherlands, 2 - 4 October 2013.

S. Vandekerckhove. Optimization of perfectly matched layers in electromagnetics using dolfin-adjoint. Oral presentation at *European Conference on Mathematics for Industry 2014.* Taormina, Italy, 9 - 13 June 2014.

S. Vandekerckhove, H. De Gersem, K. Van Den Abeele and S. Vandewalle. Simulating wave propagation problems for automated non-destructive characterization of material defects. Poster presentation at *Workshop on Advances in Electromagnetic Research 2014.* Hirshegg, Austria, 13 - 19 September 2014.

S. Vandekerckhove, H. De Gersem, K. Van Den Abeele and S. Vandewalle. Simulation of wave propagation problems for automated non-destructive characterization of material parameters and defects. Poster presentation at *Thirty-ninth Conference of the Dutch-Flemish Numerical Analysis Communities.* Zeist, The Netherlands, 8 - 10 October 2014.

S. Vandekerckhove. Automated Non-destructive evaluation using adjoint techniques. Poster presentation at *Applied inverse problems 2015.* Helsinki, Finland, May 25 - 29, 2015.

S. Vandekerckhove. Automatic non-destructive testing procedures with FEniCS and dolfin-adjoint. Oral presentation at *FEniCS 2015 Workshop.* London, United Kingdom, 29 June - 1 July 2015.

S. Vandekerckhove and H. Vandecasteele. The battle of continuous versus discontinuous finite elements to simulate elastic waves in time domain. Poster presentation at *Fortieth Conference of the Dutch-Flemish Numerical Analysis Communities.* Zeist, the Netherlands, 7 - 9 October 2015.

S. Vandekerckhove. Automated non-destructive testing and evaluation. Oral presentation at *WSC Spring meeting 2016.* Utrecht, The Netherlands, 13 May 2016.

Communications at other meetings

S. Vandekerckhove. Non-destructive Testing supported by numerical simulations. Oral presentation at *Optec Retreat 2012.* Nieuwpoort, Belgium, 8 - 9 November 2012.

S. Vandekerckhove, B. Vandewoestyne, H. De Gersem and S. Vandewalle. Numerieke simulatie van golfvoortplantingsfenomenen. Poster presentation at *Research Day Kulak.* Kortrijk, Belgium, 25 April 2012.

S. Vandekerckhove, H. De Gersem and G. N. Wells. Optimization of perfectly matched layers in electromagnetics using dolfin-adjoint. Invited presentation at *Imperial College.* London, United Kingdom, 18 July 2013.

S. Vandekerckhove, H. De Gersem, K. Van Den Abeele and S. Vandewalle. Simulation of wave propagation problems for automated non-destructive characterization of material parameters and defects. Poster presentation at *Gene Golub SIAM summer school on simulation, optimization, and identification in solid mechanics .* Linz, Austria, 4 - 15 August 2014.

S. Vandekerckhove. Simulation of wave propagation problems for automated non-destructive characterization of material parameters and defects. Oral presentation at *KU Leuven Kulak.* Kortrijk, Belgium, 13 November 2014.

S. Vandekerckhove, K. Van Den Abeele, S. Vandewalle and H. De Gersem. Toward automated methods for non-destructive testing and evaluation. Oral presentation at *Optec working group 4 meeting KU Leuven.* Leuven, Belgium, 22 April 2015.

S. Vandekerckhove, H. De Gersem, K. Van Den Abeele and S. Vandewalle. Automated identification of object properties using waves Poster presentation at *Research Day Kulak.* Kortrijk, Belgium, 13 November 2015.

S. Vandekerckhove. PDE constrained optimization with FEniCS & dolfin-adjoint Oral presentation at *Optec Retreat 2015.* Spa, Belgium, 16 - 18 November 2015.

S. Vandekerckhove. Automated non-destructive testing with FEniCS & dolfin-adjoint Invited presentation at *Lime bv.* Eindhoven, The Netherlands, 21 April 2016.

S. Vandekerckhove. Simulation of wave propagation problems for automated characterization of material parameters and defects Invited presentation at *Seminar "Numerical Methods for Field Computations and CE", Institut für Theorie Elektromagnetischer Felder (TEMF) and Graduate School of Computational Engineering, Technische Universität Darmstadt.* Darmstadt, Germany, 11 July 2016.

FACULTY OF SCIENCE
DEPARTMENT OF PHYSICS
WAVE PROPAGATION AND SIGNAL PROCESSING
E. Sabbelaan 53
B-8500 Kortrijk
steven.vandekerckhove@kuleuven.be
http://www.kuleuven-kulak.be